Apple Inc.

Apple macOS 14 Sonoma: FileVault Security Target

Version:	1.2
Status:	Final
Last Update:	2025-05-08
Validation Body:	NIAP
Validation ID:	VID11448
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
1.1	2025-04-11	atsec	First published version.
1.2	2025-05-08	atsec	Improved based on the comments from validators.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Apple macOS 14 Sonoma: FileVault Security Target
Version:	1.2
Status:	Final
Date:	2025-05-08
Sponsor:	Apple Inc.
Developer:	Apple Inc.
Validation Body:	NIAP
Validation ID:	VID11448
Keywords:	FileVault, Full Drive Encryption, Encryption Engine, Authorization Acquisition

1.2 TOE Identification

The TOE is Apple macOS 14 Sonoma: FileVault.

1.3 TOE Type

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

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 Apple macOS 14 Sonoma: FileVault, which is a Full Drive Encryption (FDE) solution including both Authorization Acquisition (AA) and Encryption Engine (EE) components. The TOE is a built-in security feature providing data-at-rest protection on Apple Mac computers. It is a hybrid FDE implementation based on a single vendor's combination of hardware and software.

The Mac computers run Apple macOS operating system. Apple macOS is a POSIX-compliant operating system (OS) built on top of the XNU kernel. The TOE is part of the macOS operating system which leverages the Apple silicon System on Chip (SoC) or the Apple T2 Security Chip. Included in the Apple silicon SoC and the Apple T2 Security Chip are Apple Secure Enclave and DMA Storage Controller. The Secure Enclave is a dedicated secure subsystem where all FDE cryptographic key handling occurs. The DMA Storage Controller provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between the storage and main memory, making data encryption with AES-XTS efficient. A special channel from the Secure Enclave securely transfers necessary keying material to the AES engine.

The tested version of the TOE is:

- Apple macOS 14.2.1

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 Mac computers listed in Appendix A.1 "Devices Covered by this Evaluation". These Mac computers are organized into the following two groups:

- Apple silicon Mac
- "Intel with T2" Mac

The Apple silicon Mac group represents all systems listed in Appendix A.1 that use an Apple silicon SoC. The "Intel with T2" Mac 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:

- Apple macOS 14 Sonoma: FileVault Common Criteria Configuration Guide v1.0

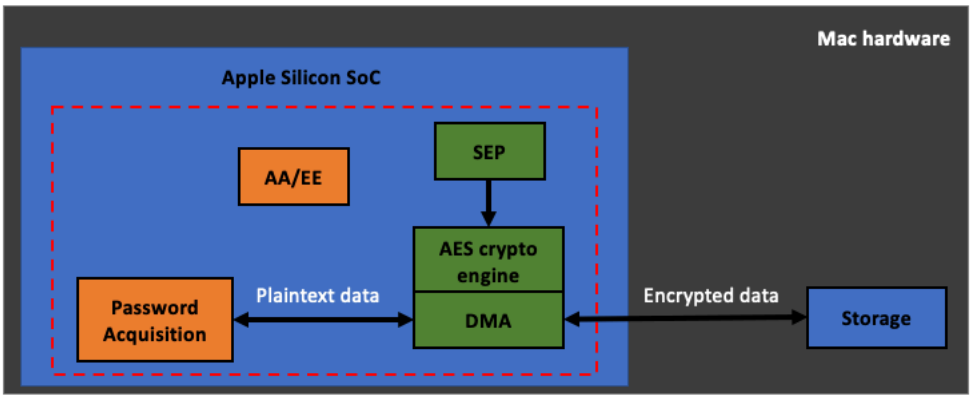
1.5.1.1 Apple silicon

The Apple silicon SoC includes

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

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

Figure 1: Apple silicon: Major components of TOE



1.5.1.2 Intel with T2

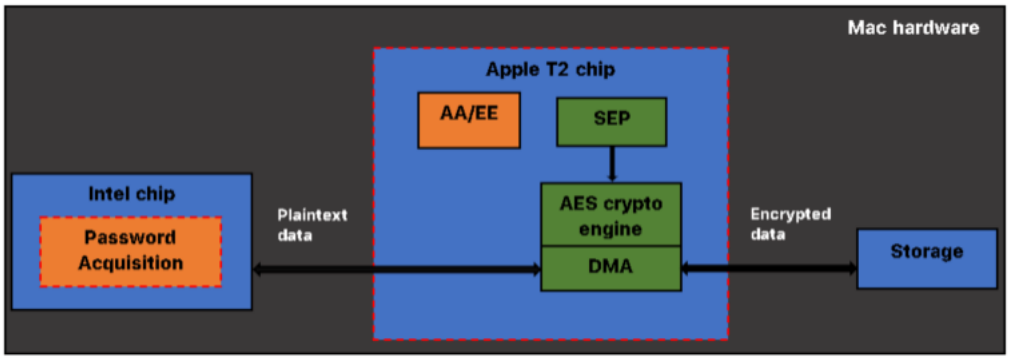
The Apple T2 Security Chip includes

- the application processor, which serves as a co-processor to the Intel processor and runs the T2OS operating system; the Intel processor is the main processor running macOS;

- the Secure Enclave, which contains the SEP running the sepOS operating system; and
- the DMA Storage Controller, which performs the storage encryption.

The EE component is instantiated on the T2. The AA component is instantiated on both the Intel processor (Password Acquisition) and the T2. The Secure Enclave provides security related functionality for EE (other than encryption/ decryption of storage data), as well as all of the cryptographic functionality for AA (i.e., PBKDF2). The DMA Storage Controller provides a dedicated AES crypto engine built into the DMA path between the storage and main memory of the host platform. The Password Acquisition function of AA component is implemented as the pre-boot component on the storage drive. It captures the user password and passes it to the Secure Enclave.

Figure 2: Intel with T2: Major components of TOE



1.5.1.3 Secure Enclave

The Secure Enclave is a dedicated secure subsystem integrated into the Apple silicon SoC and 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. The 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 TOE device;
- not accessible to other parts of the system, and
- not known to Apple.

1.5.2 TOE security functionality

The TOE provides the security functions required by the protection profiles listed in Section 2 "CC Conformance Claim".

1.5.2.1 Cryptographic support (FCS)

The TOE employs a collection of cryptographic modules to satisfy the cryptographic requirements claimed in this ST.

On Apple silicon Mac computers, the TOE uses the following cryptographic modules:

Apple corecrypto Module v14.0 [Apple silicon, User, Software, SL1]
Cryptographic library offering various cryptographic mechanisms in the macOS user space.

Apple corecrypto Module v14.0 [Apple silicon, Kernel, Software, SL1]
Cryptographic library offering various cryptographic mechanisms in the macOS kernel space.

Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
Cryptographic library offering various cryptographic mechanisms in the Secure Enclave Processor environment. It contains both firmware and hardware cryptographic algorithm implementations. Secure Key Store is also known as SKS.

Apple DMA Storage Controller v2.0 [Hardware]
The AES-XTS cryptographic algorithm implementation in the DMA Storage Controller.

On "Intel with T2" Mac computers, the TOE uses the following cryptographic modules:

Apple corecrypto Module v14.0 [Intel, User, Software, SL1]
Cryptographic library offering various cryptographic mechanisms in the macOS user space.

Apple corecrypto Module v14.0 [Intel, Kernel, Software, SL1]
Cryptographic library offering various cryptographic mechanisms in the macOS kernel space.

Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
Cryptographic library offering various cryptographic mechanisms in the Secure Enclave Processor environment. It contains both firmware and hardware cryptographic algorithm implementations. Secure Key Store is also known as SKS.

Apple DMA Storage Controller v1.0 [Hardware]
The AES-XTS cryptographic algorithm implementation in the DMA Storage Controller.

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

Table 1: Cryptographic algorithms

Algorithm	Standard
AES-CBC	NIST SP 800-38A [SP800-38A]
AES-KW	NIST SP 800-38F [SP800-38F]
AES-XTS	NIST SP 800-38E [SP800-38E]
ECDSA	FIPS 186-4 [FIPS186-4]
RSA	FIPS 186-4 [FIPS186-4]
HMAC	FIPS 198-1 [FIPS198-1]
SHA	FIPS 180-4 [FIPS180-4]
CTR_DRBG (AES)	NIST SP 800-90A Rev. 1 [SP800-90Ar1]

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 software updates using digital signatures

The macOS operating system retrieves the update package from the Apple Update Server and forwards the package to the AA component. The TOE validates the digital 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

2 CC Conformance Claim

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

This Security Target claims exact conformance to the following PP-Configuration and Protection Profiles (PPs):

- [CFG_CPP_FDE_AA-CPP_FDE_EE_V1.0]: PP-Configuration for Full Drive Encryption - Authorization Acquisition and Full Drive Encryption - Encryption Engine, Version 1.0 as of 2024-05-31, which consists of the following components:
  - [CPP_FDE_AA_V2.0E]: collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, Version 2.0 + Errata 20190201 as of 2019-02-01.
  - [CPP_FDE_EE_V2.0E]: collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201 as of 2019-02-01.

2.1 collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition [CPP_FDE_AA_V2.0E]

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 CPP_FDE_AA_V2.0E

NIAP TD	TD description	Applicable?	Non-applicability rationale
TD0901	FIT Technical Decision: Clarification to FCS_PCC_EXT.1.1	Yes	
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	
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	
TD0764	FIT Technical Decision for FCS_PCC_EXT.1	Yes	
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	Yes	
TD0759	FIT Technical Decision for FCS_AFA_EXT.1.1	No	The TOE does not include the use of smartcard.
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	

2.2 collaborative Protection Profile for Full Drive Encryption - Encryption Engine [CPP_FDE_EE_V2.0E]

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 CPP_FDE_EE_V2.0E

NIAP TD	TD description	Applicable?	Non-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	
TD0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Yes	
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	

3 Security Problem Definition

The security problem definition has been taken from [CPP_FDE_AA_V2.0E] and [CPP_FDE_EE_V2.0E] and is reproduced here for the convenience of readers.

3.1 Threat Environment

3.1.1 Threats countered by the TOE

T.UNAUTHORIZED_DATA_ACCESS

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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.

3.3 Organizational Security Policies

There are no organizational security policies addressed by the PPs.

4 Security Objectives

The security objectives have been taken from [CPP_FDE_AA_V2.0E] and [CPP_FDE_EE_V2.0E] and are reproduced here for the convenience of readers.

There are no security objectives for the TOE defined in the PPs.

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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

OE.INITIAL_DRIVE_STATE

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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

OE.POWER_DOWN/EE

PP Origin: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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

OE.PLATFORM_STATE

PP Origin: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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 is defined in the protection profiles listed in Section 2 "CC Conformance Claim".

5 Extended Components Definition

The extended components are defined in the protection profiles listed in Section 2 "CC Conformance Claim".

6 Security Requirements

6.1 TOE Security Functional Requirements

For brevity, the following convention has been used in the tables of this section to identify the source of a security requirement:

- AA: [CPP_FDE_AA_V2.0E]
- EE: [CPP_FDE_EE_V2.0E]
- AA_EE: [CPP_FDE_AA_V2.0E] and [CPP_FDE_EE_V2.0E]
- CC: Common Criteria [CC] version 3.1 revision 5

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

Security functional class	Security functional requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	FCS_COP.1(c)/AA Cryptographic Operation (Keyed Hash Algorithm)	AA	No	No	Yes	Yes
	FCS_COP.1(c)/EE Cryptographic Operation (Message Authentication)	EE	No	No	Yes	Yes
	FCS_COP.1(d) Cryptographic Operation (Key Wrapping)	AA_EE	No	No	No	Yes
	FCS_COP.1(f) Cryptographic Operation (AES Data Encryption/Decryption)	EE	No	No	No	Yes
	FCS_COP.1(g) Cryptographic Operation (Key Encryption)	AA	No	No	No	Yes
	FCS_KYC_EXT.1 Key Chaining (Initiator)	AA	No	Yes	Yes	Yes
	FCS_KYC_EXT.2 Key Chaining (Recipient)	EE	No	No	No	Yes
	FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning	AA	No	No	Yes	Yes
	FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)	AA_EE	No	No	Yes	Yes
	FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)	AA_EE	No	No	No	Yes
	FCS_VAL_EXT.1/AA Validation	AA	No	Yes	Yes	Yes
	FCS_VAL_EXT.1/EE Validation	EE	No	Yes	Yes	Yes
FDP - User data protection	FDP_DSK_EXT.1 Protection of Data on Disk	EE	No	No	No	No
FMT - Security management	FMT_MOF.1 Management of Functions Behavior	AA	No	No	No	No
	FMT_SMF.1/AA Specification of Management Functions - Authorization Acquisition	AA	No	No	No	Yes
	FMT_SMF.1/EE Specification of Management Functions - Encryption Engine	EE	No	Yes	No	Yes
	FMT_SMR.1 Security Roles - Authorization Acquisition	AA	No	No	No	No
FPT - Protection of the TSF	FPT_FUA_EXT.1 Firmware Update Authentication	EE	No	Yes	No	Yes
	FPT_KYP_EXT.1/AA Protection of Key and Key Material (AA)	AA	No	No	No	Yes
	FPT_KYP_EXT.1/EE Protection of Key and Key Material (EE)	EE	No	No	No	Yes
	FPT_PWR_EXT.1/AA Power Saving States (AA)	AA	No	No	No	Yes
	FPT_PWR_EXT.1/EE Power Saving States (EE)	EE	No	No	No	Yes
	FPT_PWR_EXT.2 Timing of Power Saving States	AA_EE	No	Yes	No	Yes
	FPT_TST_EXT.1 Testing	AA_EE	No	No	Yes	Yes
	FPT_TUD_EXT.1/AA Trusted Update	AA	No	No	No	Yes
	FPT_TUD_EXT.1/EE Trusted Update	EE	No	No	No	Yes

6.1.1 Cryptographic support (FCS)

6.1.1.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

PP Origin: *CPP_FDE_AA_V2.0E*

FCS_AFA_EXT.1.1

The TSF shall accept the following authorization factors:

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

TSS Link: *TSS for FCS_AFA_EXT.1*

6.1.1.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

PP Origin: *CPP_FDE_AA_V2.0E*

FCS_AFA_EXT.2.1

The TSF shall reacquire the authorization factor(s) specified in FCS_AFA_EXT.1 upon transition from any Compliant power saving state specified in FPT_PWR_EXT.1/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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_CKM.1.1(b)

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

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

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

PP Origin: *CPP_FDE_EE_V2.0E*

FCS_CKM.1.1(c)

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation 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: *CPP_FDE_AA_V2.0E*

FCS_CKM.4.1(a)/AA

The TSF shall **erase** cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1/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: *CPP_FDE_EE_V2.0E*

FCS_CKM.4.1(a)/EE

The TSF shall **erase** cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1/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: *CPP_FDE_EE_V2.0E*

FCS_CKM.4.1(b)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method:

- **For volatile memory, the destruction shall be executed by a**
  - **single overwrite consisting of**
    - **zeroes**
  - **removal of power to the memory**
- **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: *CPP_FDE_AA_V2.0E*

FCS_CKM.4.1(d)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

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

- **removal of power to the memory**

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_CKM_EXT.4.1(a)

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

TSS Link: *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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_CKM_EXT.4.1(b)

The TSF shall destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1/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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_COP.1.1(b)

The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm **SHA-256**, **SHA-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: *CPP_FDE_AA_V2.0E*

FCS_COP.1.1(c)/AA

The TSF shall perform cryptographic [keyed-hash message authentication] in accordance with a specified cryptographic algorithm **HMAC-SHA-256** and cryptographic key sizes **256 bits 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: *CPP_FDE_EE_V2.0E*

FCS_COP.1.1(c)/EE

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

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

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

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_COP.1.1(d)

The TSF shall perform [key wrapping] in accordance with a specified cryptographic algorithm [AES] in the following modes **KW** and the cryptographic key size **256 bits** that meet the 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: *CPP_FDE_EE_V2.0E*

FCS_COP.1.1(f)

The TSF shall perform [data encryption and decryption] in accordance with a specified cryptographic algorithm [AES used in **XTS** mode] and cryptographic key sizes **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: *CPP_FDE_AA_V2.0E*

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_KYC_EXT.1 Key Chaining (Initiator)

PP Origin: *CPP_FDE_AA_V2.0E*

FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of:

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

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

FCS_KYC_EXT.1.2

The TSF shall provide at least a **256 bit** BEV to **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*

6.1.1.20 FCS_KYC_EXT.2 Key Chaining (Recipient)

PP Origin: *CPP_FDE_EE_V2.0E*

FCS_KYC_EXT.2.1

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

FCS_KYC_EXT.2.2

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

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

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

TSS Link: *TSS for FCS_KYC_EXT.2*

6.1.1.21 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

PP Origin: *CPP_FDE_AA_V2.0E*

Applied TDs: *TD0764*, *TD0901*

FCS_PCC_EXT.1.1

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

TSS Link: *TSS for FCS_PCC_EXT.1*

6.1.1.22 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from

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

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

TSS Link: *TSS for FCS_RBG_EXT.1*

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

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

Applied TDs: *TD0760*

FCS_SNI_EXT.1.1

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

FCS_SNI_EXT.1.2

The TSF shall use **no nonces**.

FCS_SNI_EXT.1.3

The TSF shall **create IVs in the following manner**

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

TSS Link: *TSS for FCS_SNI_EXT.1*

6.1.1.24 FCS_VAL_EXT.1/AA Validation

PP Origin: *CPP_FDE_AA_V2.0E*

FCS_VAL_EXT.1.1/AA

The TSF shall perform validation of the **BEV** using the following method(s):

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

FCS_VAL_EXT.1.2/AA

The TSF shall require validation of the [BEV] prior to [forwarding the BEV to the EE].

FCS_VAL_EXT.1.3/AA

The TSF shall

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

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

6.1.1.25 FCS_VAL_EXT.1/EE Validation

PP Origin: *CPP_FDE_EE_V2.0E*

FCS_VAL_EXT.1.1/EE

The TSF shall perform validation of the [BEV] using the following method(s):

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

FCS_VAL_EXT.1.2/EE

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

FCS_VAL_EXT.1.3/EE

The TSF shall

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

TSS Link: *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: *CPP_FDE_EE_V2.0E*

FDP_DSK_EXT.1.1

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

FDP_DSK_EXT.1.2

The TSF shall encrypt all protected data without user intervention.

TSS Link: *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: *CPP_FDE_AA_V2.0E*

FMT_MOF.1.1

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

TSS Link: *TSS for FMT_MOF.1*

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

PP Origin: *CPP_FDE_AA_V2.0E*

Applied TDs: *TD0767*

FMT_SMF.1.1/AA

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

a) 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: *CPP_FDE_EE_V2.0E*

FMT_SMF.1.1/EE

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

a) change the DEK, as specified in FCS_CKM.1(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: *CPP_FDE_AA_V2.0E*

FMT_SMR.1.1

The TSF shall maintain the roles [authorized user].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

TSS Link: *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: *CPP_FDE_EE_V2.0E*

FPT_FUA_EXT.1.1

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

FPT_FUA_EXT.1.2

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

FPT_FUA_EXT.1.3

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

FPT_FUA_EXT.1.4

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

TSS Link: *TSS for FPT_FUA_EXT.1*

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

PP Origin: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E*

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: *CPP_FDE_EE_V2.0E*

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: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FPT_PWR_EXT.2.1

For each Compliant power saving state defined in *FPT_PWR_EXT.1.1/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, **shutdown**.

TSS Link: *TSS for FPT_PWR_EXT.2*

6.1.4.7 FPT_TST_EXT.1 Testing

PP Origin: *CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E*

FPT_TST_EXT.1.1

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

a) **authenticity and integrity check of software/firmware**

b) **Known Answer Tests (KATs)**

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: *CPP_FDE_AA_V2.0E*

FPT_TUD_EXT.1.1/AA

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

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: *CPP_FDE_EE_V2.0E*

FPT_TUD_EXT.1.1/EE

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

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 is defined in the protection profiles listed 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 TOE assurance requirements have been taken from [CPP_FDE_AA_V2.0E] and [CPP_FDE_EE_V2.0E].

Given the criticality of the key management scheme, a Key Management Description (KMD) document is required in the CC evaluation of Full Drive Encryption. In addition, an Entropy Essay regarding the quality of the entropy is required if the TOE includes a random bit generator. The SAR element ASE_TSS.1.1C is refined in [CPP_FDE_AA_V2.0E] and [CPP_FDE_EE_V2.0E] to specify the requirements of KMD and Entropy Essay.

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_TSS.1 TOE summary specification	AA_EE	No	No	No	Yes
	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
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

6.3.1 ASE Security Target evaluation

6.3.1.1 ASE_TSS.1 TOE summary specification

Developer action elements:

ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

Content and presentation elements:

ASE_TSS.1.1C

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

Evaluator action elements:

ASE_TSS.1.1E

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

ASE_TSS.1.2E

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

6.4 Security Assurance Requirements Rationale

The rationale is defined in the protection profiles listed 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 for SFRs

TOE SFRs	TOE summary specification
FCS_AFA_EXT.1, FCS_PCC_EXT.1 (Authorization factor, PBKDF2)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, the TOE accepts the password authorization factor from the user. 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 characters. For each user authenticating to the TOE, a 256-bit AES key, termed Password-Derived Key (PDK) in the ST, is derived using the following information: • User login password, • Per-user salt, • Secure Enclave UID (refer to section 1.5.1.3). The PDK is defined as the password submask and Border Encryption Value (BEV) in the TOE. The TOE applies the PDK as the key to unwrap a given ciphertext associated with the user account using the 256-bit AES-KW algorithm [SP800-38F]. If the decryption succeeds, the authentication is successful and the user is logged in to the system. Considering the self-authenticating feature of AES-KW algorithm, a failed unwrapping operation implies that the user's password must be wrong, as all other parts used to derive the PDK remain unchanged. The TOE implements PBKDF2 in the Secure Enclave to derive a 256-bit 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 password 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 loop is performed as often as needed to reach a duration between 100 and 150 milliseconds on the TOE device, with a minimum of 50,000 iterations. The final output, after all AES iterations have completed, forms the 256-bit PDK.
FCS_AFA_EXT.2 (Authorization factor acquisition)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, 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) (Symmetric key generation)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, at the time of creating a data volume, the TOE uses the random bit generator specified in FCS_RBG_EXT.1 to generates the following 256-bit AES keys: • A Volume Encryption Key (VEK) for encrypting the volume. • A Key Encryption Key (KEK) for protecting the VEK.

TOE SFRs	TOE summary specification
	Volume and metadata contents are encrypted with the VEK, which is wrapped with the KEK.
FCS_CKM.1(c) (Data encryption key generation)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, when creating a data volume, the TOE uses the random bit generator specified in FCS_RBG_EXT.1 to generates a 256-bit AES key, which is termed Volume Encryption Key (VEK). This key serves as the Data Encryption Key (DEK) for encrypting the data volume.
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" Mac computers, the TOE leverages NAND flash for non-volatile memory, and leverages DRAM for volatile memory. In NAND flash, reading and writing data occur at the page level. To protect data from hardware errors, the flash drive associates each page with an error correcting code (ECC) checksum. In DRAM, data is read and written at the byte level without redundant bits. All symmetric keys stored in non-volatile memory are cryptographically wrapped. The TOE erases cryptographic keys and key material from non-volatile memory by performing a single overwrite of zeroes. Keys are introduced into volatile memory only when they are required to perform a specific cryptographic operation. Since the keys are being used by the Secure Enclave to perform the operation, the Secure Enclave tracks the memory location of the key until the operation is complete. Once a key is no longer needed for the specific operation, the key is erased from volatile memory. The TOE erases cryptographic keys and key material from volatile memory by performing a single overwrite of zeroes and/or by removal of power to the memory. The erase operation is performed by the Secure Enclave 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). The Secure Enclave UID is fused into the Secure Enclave. The UID is not accessible by any component outside of the Secure Enclave and cannot be erased. The TOE will destroy all key material and cryptographic keys stored in plaintext when transitioning to a Compliant power saving state as defined in FPT_PWR_EXT.1/AA and FPT_PWR_EXT.1/EE.

FCS_COP.1(a) (Signature verification)	**Summary**
	
	On both Apple silicon and "Intel with T2" Mac computers, 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: [FIPS186-4]
	Modules:
	• Apple corecrypto Module v14.0 [Apple silicon, User, Software, SL1]
	• Apple corecrypto Module v14.0 [Apple silicon, Kernel, Software, SL1]

TOE SFRs	TOE summary specification
	• Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] On Apple silicon Mac computers, signatures are verified using ECDSA P-521 and SHA-512. The CA public key is embedded in the 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:* [FIPS186-4] *Modules:* • Apple corecrypto Module v14.0 [Intel, User, Software, SL1] • Apple corecrypto Module v14.0 [Intel, Kernel, Software, SL1] • Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] On "Intel with T2" Mac computers, signatures are verified using RSA 4096-bit and SHA-256. The CA public key is embedded in the 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:* [FIPS180-4] *Modules:* • Apple corecrypto Module v14.0 [Apple silicon, User, Software, SL1] • Apple corecrypto Module v14.0 [Apple silicon, Kernel, Software, SL1] • Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] **Intel with T2** *Algorithm:* SHA-256 *Standard:* [FIPS180-4] *Modules:* • Apple corecrypto Module v14.0 [Intel, User, Software, SL1] • Apple corecrypto Module v14.0 [Intel, Kernel, Software, SL1] • Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
FCS_COP.1(c)/AA, FCS_COP.1(c)/EE (Keyed-hash message authentication)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, 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 following parameters are used in the HMAC-SHA-256 algorithm: • Key length: 256 bits • Block size: 512 bits • Output MAC length: 256 bits *Algorithm:* HMAC-SHA-256 *Standard:* [FIPS198-1] **Apple silicon** *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]

TOE SFRs	TOE summary specification
	Intel with T2 *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
FCS_COP.1(d) (Key wrapping)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, 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 password is able to successfully unwrap a given ciphertext associated with the user account. The failure of unwrapping operation is a user authentication failure and therefore access will be denied. *Algorithm:* AES-KW-256 *Standard:* [SP800-38F] **Apple silicon** *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] **Intel with T2** *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
FCS_COP.1(f) (Data encryption and decryption)	**Summary** The TOE uses the AES-XTS mode for data encryption and decryption. The DEK is generated as per the TSS for FCS_CKM.1(c). **Apple silicon** *Algorithm:* AES-XTS-256 *Standard:* [SP800-38E] *Module:* Apple DMA Storage Controller v2.0 [Hardware] On Apple silicon Mac computers, 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** *Algorithm:* AES-XTS-128 *Standard:* [SP800-38E] *Module:* Apple DMA Storage Controller v1.0 [Hardware] On "Intel with T2" Mac computers, 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_COP.1(g) (Key encryption)	**Summary** The TOE uses the hardware AES-CBC-256 key encryption implementation when generating the PDK as described in the TSS for FCS_PCC_EXT.1. The key size supported is 256 bits.

TOE SFRs	TOE summary specification
	Algorithm: AES-CBC-256 *Standard:* [SP800-38A] **Apple silicon** *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] **Intel with T2** *Module:* Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]
FCS_KYC_EXT.1, FCS_KYC_EXT.2 (Key chaining initiator & recipient)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, the TOE supports BEV sizes of 256 bits. As a key chaining initiator, the TOE maintains a key chain of one key, using a 256-bit password submask as the BEV. The password conditioning process is described in the TSS for FCS_PCC_EXT.1 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" Mac computers, the TOE performs deterministic random bit generation services according to [SP800-90Ar1] using CTR_DRBG(AES). The DRBG is implemented in hardware as part of the Secure Enclave TRNG (part of the TOE hardware).
FCS_SNI_EXT.1 (Salt, nonce, and IV generation)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, the TOE generates salts and initialization vectors (IVs) using the Secure Enclave DRBG. • Salts are 16 bytes and are used with the PBKDF2. • The AES-CBC IVs are used when deriving the PDK from password. 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** Since the TOE consists of both FDE AA and FDE EE components, the BEV validation processes in both AA and EE components are implemented as a single process in the TOE. On both Apple silicon and "Intel with T2" Mac computers, the TOE validates a BEV using the 256-bit AES-KW algorithm. The TOE applies the BEV as the key to unwrap a given ciphertext associated with the user account. If the decryption succeeds, the BEV is regarded as valid. Otherwise, the BEV is invalid. 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	**Summary**

TOE SFRs	TOE summary specification
(Protection of data on disk)	On both Apple silicon and "Intel with T2" Mac computers, the TOE provides a dedicated AES-XTS crypto engine built into the DMA path between the flash storage and the main memory of the host platform. This DMA Storage Controller is placed in the middle of the data path between the application processor and the storage device. The DMA Storage Controller performs the encryption/ decryption of the data prior to reaching the application processor or the storage. When a read operation is made, the data must first be decrypted by the DMA Storage Controller before the application processor has access to the data. When a write operation is made, the data is first encrypted by the DMA Storage Controller and then written to storage as a block of encrypted data. This arrangement ensures that standard methods of accessing the storage drive via the operating system will pass through these functions. 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), system volumes (protected by the signed system volume feature), 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 on macOS. 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.
FMT_MOF.1 (Function behavior management)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, the TOE supports only one Compliant power saving state: • G2(S5) - soft off state: the system is powered down. This behaviour cannot be modified by users.

FMT_SMF.1/AA, FMT_SMF.1/EE (Management functions)	**Summary**
	
	On both Apple silicon and "Intel with T2" Mac computers, the TOE supports the following management functions:
	• Authorization Acquisition:
	  ○ Forwarding requests to change the DEK to the EE:
	  ➢ Since the TOE consists of both FDE AA and FDE EE components, the requests to change the DEK are processed by the EE component directly.
	  ○ Forwarding requests to cryptographically erase the DEK to the EE:

TOE SFRs	TOE summary specification
	▸ Since the TOE consists of both FDE AA and FDE EE components, the requests to erase the DEK are processed by the EE component directly. ○ 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 can check for and install updates to the TOE via either the System Settings app or the `softwareupdate` command on macOS. When creating a data volume, the TOE randomly generate a Data Encryption Key (DEK) and a Key Encryption Key (KEK). The DEK is protected by wrapping it with the KEK. Furthermore, the KEK is wrapped with the Password-Derived Key (PDK), thus providing data confidentiality based on passwords. The DEK and KEK are stored as part of the volume's metadata. The PDK is ephemerally stored in a dedicated region of the TOE device's memory, which is protected by the Secure Enclave. Multiple layers of protection isolate the Secure Enclave protected memory from the Application Processor. The PDK is deleted right after the user authentication process has been completed and the KEK has been unwrapped. The wrapped DEK is further wrapped using the media key, which is stored in the Effaceable Storage of the Secure Enclave to facilitate fast wipe rather than confidentiality. The media key is generated during system installation, or re-generated on system wipe. When deleting a volume, its DEK is securely deleted by the Secure Enclave. The media key is designed to be quickly erased on demand; for example, via remote wipe using Find My Mac or when enrolled in a mobile device management (MDM) solution. Effaceable storage accesses the underlying storage technology to directly address and erase a small number of blocks at a very low level. Once the media key is deleted by the Secure Enclave, the DEK will become cryptographically inaccessible and, therefore, the volume will be cryptographically inaccessible.
FMT_SMR.1 (Security roles)	**Summary** On both Apple silicon and "Intel with T2" Mac computers, the TOE supports authorized user role and it can associate users to roles.
FPT_FUA_EXT.1,	**Summary**

TOE SFRs	TOE summary specification
FPT_TUD_EXT.1/AA, FPT_TUD_EXT.1/EE (Software/firmware updates)	Apple Update Server is leveraged for downloading software/firmware update packages. The update package contains the TOE software, T2OS/firmware (for "Intel with T2" Mac computers only), and sepOS/firmware. All of them are bundled into the macOS update package. The TOE stores the download in a temporary location on flash. Once the download is complete, the TOE verifies the digital signature of the update package 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 TOE device. If the verification fails, the TOE terminates the update process with an error message. The user can check for and install updates to the TOE via either the System Settings app or the `softwareupdate` command on macOS.
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" Mac computers, the TOE supports the following Compliant power saving state: • G2(S5) - soft off state: the system is powered down. For the TOE to enter the Compliant power saving state, the user should either select the menu option Apple menu » Shut Down, or press and hold the physical power button .
FPT_TST_EXT.1 (Testing)	**Summary** **Secure boot** When the TOE device is turned on, the boot process starts with executing code from read-only memory referred to as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication and is implicitly trusted. Each following step of the boot process contains components that are cryptographically signed by Apple to enable integrity checking. The boot process proceeds only after verifying the integrity of the software at every step, which creates a chain of trust rooted in hardware. • Apple silicon: When an Apple silicon Mac is turned on, the application processor executes code from the Boot ROM in the first step in the chain of trust. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the Low-Level Bootloader (LLB) is signed by Apple's private key before allowing it to load. When LLB is launched, it verifies and loads the iBoot bootloader. When the iBoot finishes its tasks, it verifies and runs the macOS kernel. • Intel with T2: When an "Intel with T2" Mac in turned on, the Apple T2 Security Chip performs a secure boot from its Boot ROM in the same fashion as a Mac with Apple silicon. It executes the Boot ROM code to verifies and loads the iBoot bootloader. The iBoot bootloader verifies the kernel and kernel extension code on the T2 chip, which subsequently verifies the Intel Unified Extensible Firmware Interface (UEFI) firmware. After verification, the UEFI firmware image is mapped into a portion of the T2 chip memory and this memory is made available to the Intel processor. The boot process continues on the Intel processor, with the UEFI firmware evaluating the signature for boot.efi, which is the macOS bootloader. The boot.efi code in turn verifies and load the macOS kernel. **Self-tests** Self-tests are performed by the cryptographic modules included in the TOE. The FIPS Self-Tests application runs all required module self-tests. This application is invoked by the TOE OS startup process when the TOE device powers on. At start-up, the cryptographic modules perform Known Answer Tests (KATs) to ensure the correctness of the cryptographic functions. • CTR_DRBG: health testing implemented as KAT • HMAC-SHA-256: KAT

TOE SFRs	TOE summary specification

	• AES-CBC: KAT
	• Apple silicon:
	  ○ AES-XTS-256: KAT
	  ○ ECDSA SigVer: KAT
	• Intel with T2:
	  ○ AES-XTS-128: KAT
	  ○ RSA SigVer: KAT

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

CAVP
Cryptographic Algorithm Validation Program

CBC
Cipher 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

DRAM
Dynamic Random-Access Memory

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

KMD
Key Management Description

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

PDK
Password-Derived Key

PP
Protection Profile

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 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

VEK
Volume Encryption Key

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

CFG_CPP_FDE_AA-CPP_FDE_EE_V1.0 — **PP-Configuration for Full Drive Encryption - Authorization Acquisition and Full Drive Encryption - Encryption Engine**
Version 1.0
Date 2024-05-31
Location https://www.niap-ccevs.org/protectionprofiles/437

CPP_FDE_AA_V2.0E — **collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition**
Version 2.0 + Errata 20190201
Date 2019-02-01
Location https://www.niap-ccevs.org/protectionprofiles/437

CPP_FDE_EE_V2.0E — **collaborative Protection Profile for Full Drive Encryption - Encryption Engine**
Version 2.0 + Errata 20190201
Date 2019-02-01
Location https://www.niap-ccevs.org/protectionprofiles/438

FIPS180-4 — **Secure Hash Standard (SHS)**
Date 2015-08-04
Location https://csrc.nist.gov/pubs/fips/180-4/upd1/final

FIPS186-4 — **Digital Signature Standard (DSS)**
Date 2013-07-19
Location https://csrc.nist.gov/pubs/fips/186-4/final

FIPS198-1 — **The Keyed-Hash Message Authentication Code (HMAC)**

	Date 2008-07-16 Location https://csrc.nist.gov/pubs/fips/198-1/final
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
SP800-38A	**Recommendation for Block Cipher Modes of Operation: Methods and Techniques** Date 2001-12-01 Location https://csrc.nist.gov/pubs/sp/800/38/a/final
SP800-38E	**Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices** Date 2010-01-18 Location https://csrc.nist.gov/pubs/sp/800/38/e/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
SP800-90Ar1	**Recommendation for Random Number Generation Using Deterministic Random Bit Generators** Date 2015-06-24 Location https://csrc.nist.gov/pubs/sp/800/90/a/r1/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 (14-inch, Nov 2023)	A2992	Mac15,10	M3 Max	ARMv8.6-A	SEP v2.0
		Mac15,8	M3 Max	ARMv8.6-A	SEP v2.0
		Mac15,6	M3 Pro	ARMv8.6-A	SEP v2.0
	A2918	Mac15,3	M3	ARMv8.6-A	SEP v2.0
MacBook Pro (16-inch, Nov 2023)	A2991	Mac15,11	M3 Max	ARMv8.6-A	SEP v2.0
		Mac15,9	M3 Max	ARMv8.6-A	SEP v2.0
		Mac15,7	M3 Pro	ARMv8.6-A	SEP v2.0
iMac (24-inch, 2023, Two ports)	A2874	Mac15,4	M3	ARMv8.6-A	SEP v2.0
iMac (24-inch, 2023, Four ports)	A2873	Mac15,5	M3	ARMv8.6-A	SEP v2.0
Mac Studio (2023)	A2901	Mac14,14	M2 Ultra	ARMv8.6-A	SEP v2.0
		Mac14,13	M2 Max	ARMv8.6-A	SEP v2.0
Mac Pro (2023)	A2786	Mac14,8	M2 Ultra	ARMv8.6-A	SEP v2.0
Mac Pro (Rack, 2023)	A2787	Mac14,8	M2 Ultra	ARMv8.6-A	SEP v2.0
MacBook Air (15-inch, M2, 2023)	A2941	Mac14,15	M2	ARMv8.6-A	SEP v2.0
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)	A2681	Mac14,2	M2	ARMv8.6-A	SEP v2.0

Marketing Name	Model #	Model Identifier	SoC/Processor	MicroArch	Security Chip
Mac Studio (2022)	A2615	Mac13,2	M1 Ultra	ARMv8.5-A	SEP v2.0
		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
MacBook Air (Retina, 13-inch, 2020)	A2179	MacBookAir9,1	Core i7-1060NG7	Ice Lake	T2
MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports)	A2251	MacBookPro16,2	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	Core i5-10500 Core i5-10600 Core i7-10700K Core i9-10910	Comet Lake	T2
		iMac20,2	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, Two Thunderbolt 3 ports)	A2159	MacBookPro15,4	Core i5-8257U Core i7-8557U	Coffee Lake	T2
MacBook Pro (16-inch, 2019)	A2141	MacBookPro16,1	Core i7-9750H Core i9-9880H Core i9-9980HK	Coffee Lake	T2
		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

Marketing Name	Model #	Model Identifier	SoC/Processor	MicroArch	Security Chip
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

A.2 SFR to CAVP Mapping

The CAVP certificates contain several different SoCs and micro-architectures in the operating environment. 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 this section to identify the cryptographic modules:

USR

Apple corecrypto Module v14.0 [Apple silicon, User, Software, SL1]
The user space software module v14.0 on Apple silicon Mac computers.

Apple corecrypto Module v14.0 [Intel, User, Software, SL1]
The user space software module v14.0 on "Intel with T2" Mac computers.

KRN

Apple corecrypto Module v14.0 [Apple silicon, Kernel, Software, SL1]
The kernel space software module v14.0 on Apple silicon Mac computers.

Apple corecrypto Module v14.0 [Intel, Kernel, Software, SL1]
The kernel space software module v14.0 on "Intel with T2" Mac computers.

SKS-FW

Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] (firmware)
The Secure Key Store (SKS) firmware module v14.0 on both Apple silicon and "Intel with T2" Mac computers.

Note: *The Apple T2 Security Chip runs T2OS 14 operating system.*

SKS-HW

Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] (hardware)
The Secure Key Store (SKS) hardware module v2.0 on both Apple silicon and "Intel with T2" Mac computers.

DMA

Apple DMA Storage Controller v2.0 [Hardware]
DMA module on Apple silicon Mac computers.

Apple DMA Storage Controller v1.0 [Hardware]
DMA module on "Intel with T2" Mac computers.

Note: *The DMA modules of the TOE cannot be tested through the CAVP as of the time of evaluation, therefore a compliance test accepted by NIAP has been used for verifying the correctness of the AES-XTS algorithm implementation.*

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

Table 9: Mapping of SFRs to CAVP certificates (USR cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP
FCS_COP.1(a)	ECDSA SigVer (Apple silicon)	Curve: P-521 Hash: SHA2-512	[FIPS186-4]	A5988
	RSA SigVer (Intel)	Modulus: 4096 bits Hash: SHA2-256 Padding: PKCS#1 v1.5 and PSS	[FIPS186-4]	A6192
FCS_COP.1(b)	SHA2-512 (Apple silicon)	Byte-oriented mode	[FIPS180-4]	A5988

SFR	Algorithm	Capabilities	Standard	CAVP
	SHA2-256 (Intel)	Byte-oriented mode	[FIPS180-4]	A6198

Table 10: Mapping of SFRs to CAVP certificates (KRN cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP
FCS_COP.1(a)	ECDSA SigVer (Apple silicon)	Curve: P-521 Hash: SHA2-512	[FIPS186-4]	A5949
	RSA SigVer (Intel)	Modulus: 4096 bits Hash: SHA2-256 Padding: PKCS#1 v1.5 and PSS	[FIPS186-4]	A6201
FCS_COP.1(b)	SHA2-512 (Apple silicon)	Byte-oriented mode	[FIPS180-4]	A5949
	SHA2-256 (Intel)	Byte-oriented mode	[FIPS180-4]	A6207

Table 11: Mapping of SFRs to CAVP certificates (SKS-FW cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP
FCS_COP.1(a)	ECDSA SigVer (Apple silicon)	Curve: P-521 Hash: SHA2-512	[FIPS186-4]	A5981
	RSA SigVer (T2)	Modulus: 4096 bits Hash: SHA2-256 Padding: PKCS#1 v1.5 and PSS	[FIPS186-4]	A5981
FCS_COP.1(b)	SHA2-512 (Apple silicon)	Byte-oriented mode	[FIPS180-4]	A5981
	SHA2-256 (T2)	Byte-oriented mode	[FIPS180-4]	A5981
FCS_COP.1(c)/AA, FCS_COP.1(c)/EE	HMAC-SHA2-256 (Apple silicon)	N/A	[FIPS198-1]	A5982
	HMAC-SHA2-256 (T2)	N/A	[FIPS198-1]	A5981
FCS_COP.1(d)	AES-KW	256 bits encrypt, decrypt	[SP800-38F]	A5976

Table 12: Mapping of SFRs to CAVP certificates (SKS-HW cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP
FCS_COP.1(g)	AES-CBC	256 bits encrypt	[SP800-38A]	C330, A1469, A3496, A6547
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	DRBG 2029, A1362, A3490, A6548

Table 13: Mapping of SFRs to CAVP certificates (DMA cryptographic module)

SFR	Algorithm	Capabilities	Standard	CAVP
FCS_COP.1(f)	AES-XTS (Apple silicon)	256 bits encrypt, decrypt	[SP800-38E]	CCTL Tested
	AES-XTS (Intel)	128 bits encrypt, decrypt	[SP800-38E]	CCTL Tested

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

Table 14: Coverage of CAVP certificates for Apple silicon SoCs

SoC	Micro Architecture	USR		KRN		SKS-FW				SKS-HW	
		ECDSA SigVer	SHA-512	ECDSA SigVer	SHA-512	ECDSA SigVer	SHA-512	HMAC SHA-256	AES-KW	AES-CBC	CTR_DRBG
M1	ARMv8.5-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A1469	A1362
M1 Pro	ARMv8.5-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M1 Max	ARMv8.5-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M1 Ultra	ARMv8.5-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M2	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M2 Pro	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M2 Max	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A3496	A3490
M2 Ultra	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A6547	A6548
M3	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A6547	A6548
M3 Pro	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A6547	A6548
M3 Max	ARMv8.6-A	A5988	A5988	A5949	A5949	A5981	A5981	A5982	A5976	A6547	A6548

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 15: 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-3223	W	Cascade Lake	A6192	A6198	A6201	A6207	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-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	A6192	A6198	A6201	A6207	Tested
Intel Core i5-8257U	8th	Coffee Lake	A6192	A6198	A6201	A6207	Tested
Intel Core i7-8557U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i7-9750H	9th	Coffee Lake					Intel Core i9-9880H
Intel Core i9-9880H	9th	Coffee Lake	A6192	A6198	A6201	A6207	Tested
Intel Core i9-9980HK	9th	Coffee Lake					Intel Core i9-9880H
Intel Core i5-10500	10th	Comet Lake	A6192	A6198	A6201	A6207	Tested
Intel Core i5-10600	10th	Comet Lake					Intel Core i5-10500
Intel Core i7-10700K	10th	Comet Lake	A6192	A6198	A6201	A6207	Tested
Intel Core i9-10910	10th	Comet Lake	A6192	A6198	A6201	A6207	Tested
Intel Core i7-1060NG7	10th	Ice Lake	A6192	A6198	A6201	A6207	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 16: Coverage of CAVP certificates for Apple T2 Security Chip

SoC	Micro Architecture	SKS-FW				SKS-HW	
		RSA SigVer	SHA-256	HMAC SHA-256	AES-KW	AES-CBC	CTR_DRBG
T2	ARMv8.1-A	A5981	A5981	A5981	A5976	C330	DRBG 2029