Apple FileVault 2 on T2 systems running macOS Catalina 10.15 Security Target

Prepared By:
Acumen Security, LLC
www.acumensecurity.net

Prepared for: Apple
One Apple Park Way
Cupertino, CA 95014

Document Version: 2.5
Date: April 19, 2021

Table of Contents

Revision History

Version	Date	Description
0.1	10/23/2019	Initial Draft
0.2	10/24/2019	Completion of Sections 4, and 5
0.3	10/25/2019	Section 6
0.4	10/30/2019	Section 6 completion and begin SARs
0.5	10/31/2019	Began Section 7
0.6	11/04/2019	Updating Section 7 TSS
0.7	11/06/2019	Completion of TSS Section
0.8	12/2/2019	Updates made as a result of testing
0.9	2/3/2020	Updates made based on vendor comments and processor specifications and CAVP.
1.0	5/11/2020	Updates made to the platforms.
1.1	9/8/2020	Updated TOE version, and minor changes
1.2	9/29/2020	Updated TOE power savings state based on vendor feedback, updated TOE software and firmware versions, replaced the term “Master Key” with “Unlock Key” based on vendor feedback along with other minor updates.
1.3	10/13/2020	Revised description for TOE software/firmware update in the TSS.
1.4	11/06/2020	Updated CAVP Certificate table, T2 SEP description, fixed TBDs.
1.5	11/19/2020	Updating ST based on ST evaluation feedback
1.6	11/19/2020	Updated ST by removing smart card authorization factor.
1.7	11/23/2020	Updated TSS for FCS_CKM.1(c) and FCS_COP.1(d) based on Vendor feedback.
1.8	01/08/2021	Updated TSS based on CCTL internal feedback.
1.9	01/20/2021	Updated Key Management Description
2.0	02/25/2021	Inserted new diagram in KMD, updated the TOE cryptographic keys table.
2.1	03/08/2021	Updated the CAVP certificate numbers for AES-XTS and AES-KW and KMD section.
2.2	03/22/2021	Removed redundant information related to TOE cryptographic keychain, updated TSS, corrected Section 3 and addressed formatting issues.
2.3	03/25/2021	Updated based on vendor feedback.
2.4	4/16/2021	Addressing validator comments.
2.5	4/19/2021	Updated ST based on vendor feedback.

Trademarks:

Apple’s trademarks applicable to this document are listed in https://www.apple.com/legal/intellectual-property/trademark/appletmlist.html. Other company, product, and service names may be trademarks or service marks of others.

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	Apple FileVault 2 on T2 systems running macOS Catalina 10.15 Security Target
ST Version	2.5
ST Date	April 19, 2021
ST Author	Acumen Security, LLC.
TOE Identifier	Apple FileVault 2 on T2 systems running macOS Catalina 10.15
TOE Software Version	10.15.7
TOE Developer	Apple Inc.
Key Words	Full Drive Encryption, Encryption Engine, Authorization and Acquisition

Table 1: TOE and ST Identification

1.2 TOE Overview

The TOE is a full drive encryption product which supports authorization acquisition and encryption engine. The TOE is Unix-based operating system which leverages Apple T2 security processor to perform the full disk encryption. The operating system core is a POSIX compliant operating system built on top of the XNU kernel with standard Unix facilities available from the command line interface.

1.2.1 TOE Product Type

The TOE type is an authorization and encryption engine product. It satisfies all of the criterion to meet the collaborative Protection Profile for Full Drive Encryption – Encryption Engine Version 2.0 + Errata 20190201 [FDE EE v2.0e] and collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition Version 2.0 + Errata 20190201 [FDE AA v2.0e].

1.3 TOE Description

1.3.1 Evaluated Configuration

The TOE is comprised of both software and hardware. The TOE hardware consists of the Apple T2 Security Chip which is a custom silicon for the Mac. It contains the Secure Enclave coprocessor which provides security related functionality for all the EE functionality (i.e., other than encryption/decryption of storage data) and all of the cryptographic functionality for AA (i.e., PBKDF2). The Password Acquisition component (AA) is the pre-boot component on the disk and captures the user password and passes it to the T2/SEP. The T2 provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between the storage and main memory of the host platform. The T2 chip is placed in the data path between the Intel chip and the storage, enabling it to encrypt/decrypt all data flowing between these two components.

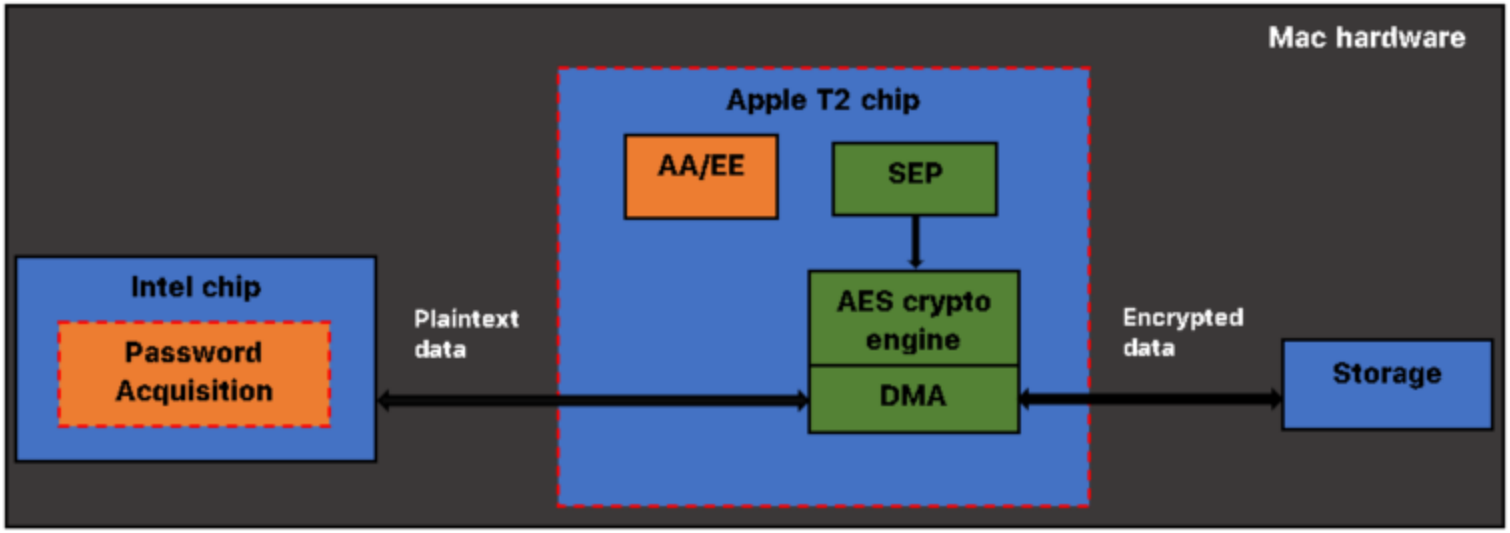

Figure 1: Major components of TOE within red border

The TOE also supports secure connectivity with an Apple update server as described in Table 2 below:

Sr. No	Component	Required	Usage/Purpose Description for TOE performance
1	Apple update server	Yes	Provides the ability to download authentic signed updates.

Table 2: IT Environment Components

Table 3 below provides a list of supported platforms:

Device	Year	Intel Processor	Apple T2 Chip
iMac Pro Model: A1862 Reference: iMac Pro1,1	Late 2017	Intel Xeon W-2140B (Skylake)	Apple T2 (ARM64) Processor T2 (processor family arm64) from Apple family: arm64 manufacturer: Apple series: T Series Software: TxFW 10.15
iMac Pro Model: A1862 Reference: iMac Pro1,1	Late 2017	Intel Xeon W-2150B (Skylake)	
iMac Pro Model: A1862 Reference: iMac Pro1,1	Late 2017	Intel Xeon W-2170B (Skylake)	
iMac Pro Model: A1862 Reference: iMac Pro1,1	Late 2017	Intel Xeon W-2191B (Skylake)	
Mac mini Model: A1993 Reference: Macmini8,1	2018	Intel Core i5-8500B (Coffee Lake)	
Mac mini Model: A1993 Reference: Macmini8,1	2018	Intel Core i7-8700B (Coffee Lake)	
MacBook Pro Model: A1989 Reference: MacBookPro15,2	Mid 2018	Intel Core i5-8279U (Coffee Lake)	
MacBook Pro Model: 1989 Reference: MacBookPro15,2	Mid 2018	Intel Core i5-8259U (Coffee Lake)	
MacBook Pro Model: A1990 Reference: MacBookPro15,1	Mid 2018	Intel Core i7-8750H (Coffee Lake)	
MacBook Pro Model: A1989 Reference: MacBookPro15,2	Mid 2018	Intel Core i7-8559U (Coffee Lake)	
MacBook Pro Model: A1990 Reference: MacBookPro15,3	Mid 2018	Intel Core i7-8850H (Coffee Lake)	

Device	Year	Intel Processor	Apple T2 Chip
MacBook Pro Model: A1990 Reference: MacBookPro15,1	Mid 2018	Intel Core i9-8950HK (Coffee Lake)	Apple T2 (ARM64) Processor T2 (processor family arm64) from Apple family: arm64 manufacturer: Apple series: T Series Software: TxFW 10.15
MacBook Pro Model: A1990 Reference: MacBookPro15,3	Mid 2018	Intel Core i9-8950HK (Coffee Lake)	
MacBook Air Model: A1932 Reference: MacBookAir8,1	Late 2018	Intel Core i5-8210Y (Amber Lake)	
MacBook Air Model: A1932 Reference: MacBookAir8,2	2019	Intel Core i5-8210Y (Amber Lake)	
Mac Pro Model: A1991 Reference: Mac Pro7,1	2019	Intel Xeon W-3223 (Cascade Lake)	
Mac Pro Model: A1991 Reference: Mac Pro7,1	2019	Intel Xeon W-3235 (Cascade Lake)	
Mac Pro Model: A1991 Reference: Mac Pro7,1	2019	Intel Xeon W-3245 (Cascade Lake)	
Mac Pro Model: A1991 Reference: Mac Pro7,1	2019	Intel Xeon W-3265M (Cascade Lake)	
Mac Pro Model: A1991 Reference: Mac Pro7,1	2019	Intel Xeon W-3275M (Amber Lake)	
MacBook Pro Model: A1989 Reference: MacBookPro15,2	2019	Intel Core i5-8279U (Amber Lake)	
MacBook Pro Model: A2159 Reference: MacBookPro15,4	2019	Intel Core i5-8257U (Amber Lake)	

Device	Year	Intel Processor	Apple T2 Chip
MacBook Pro Model: A1990 Reference: MacBookPro15,1	2019	Intel Core i7-9750H (Coffee Lake)	Apple T2 (ARM64) Processor T2 (processor family arm64) from Apple family: arm64 manufacturer: Apple series: T Series Software: TxFW 10.15
MacBook Pro Model: A1989 Reference: MacBookPro15,2	2019	Intel Core i7-8569U (Coffee Lake)	
MacBook Pro Model: A2159 Reference: MacBookPro15,4	2019	Intel Core i7-8557U (Coffee Lake)	
MacBook Pro: Model: A2141 Reference: MacBookPro16,1	2019	Intel Core i7-9750H (Coffee Lake)	
MacBook Pro Model: A1990 Reference: MacBookPro15,1	2019	Intel Core i9-9880H (Coffee Lake)	
MacBook Pro Model: A1990 Reference: MacBookPro15,1	2019	Intel Core i9-9980HK (Coffee Lake)	
MacBook Pro Model: A1990 Reference: MacBookPro15,3	2019	Intel Core i9-9880H (Coffee Lake)	
MacBook Pro Model: A2141 Reference: MacBookPro16,1	2019	Intel Core i9-9880H (Coffee Lake)	Apple T2(ARM 64) Processor T2 (processor family arm64) from Apple family: arm64 manufacturer: Apple series: T Series Software: TxFW 10.15
MacBook Pro Model: A2141 Reference: MacBookPro16,1	2019	Intel Core i9-9980HK (Coffee Lake)	
MacBook Pro Model: A2141 Reference: MacBook Pro16,4	2019	Intel Core i7-9750H (Coffee Lake)	
MacBook Pro Model: A2141 Reference: MacBook Pro16,4	2019	Intel Core i9-9880H (Coffee Lake)	

Device	Year	Intel Processor	Apple T2 Chip
MacBook Pro Model: A2141 Reference: MacBook Pro16,4	2019	Intel Core i9-9980HK (Coffee Lake)	
iMac Model: A2115 Reference: iMac20,1	2019	Intel Core i5-10500 (Ice Lake)	
Mac Pro (rack) Model: A2304 Reference: MacPro7,1	2019	Intel Xeon W-3275M (Cascade Lake)	
Mac Pro (rack) Model: A2304 Reference: MacPro7,1	2019	Intel Xeon W-3265M (Cascade Lake)	
Mac Pro (rack) Model: A2304 Reference: MacPro7,1	2019	Intel Xeon W-3245 (Cascade Lake)	
Mac Pro (rack) Model: A2304 Reference: MacPro7,1	2019	Intel Xeon W-3235 (Cascade Lake)	
Mac Pro (rack) Model: A2304 Reference: MacPro7,1	2019	Intel Xeon W-3223 (Cascade Lake)	Apple T2(ARM 64) Processor T2 (processor family arm64) from Apple family: arm64 manufacturer: Apple series: T Series Software: TxFW 10.15
MacBook Air Model: A2179 Reference: MacBook Air9,1	2020	Intel Core i5-1030NG7 (Ice Lake)	
MacBook Air Model: A2179 Reference: MacBook Air9,1	2020	Intel Core i7-1060NG7 (Ice Lake)	
MacBook Pro Model: A2289 Reference: MacBook Pro16,3	2020	Intel Core i5-8257U (Coffee Lake)	
MacBook Pro Model: A2289 Reference: MacBook Pro16,3	2020	Intel Core i7-8557U (Coffee Lake)	

Device	Year	Intel Processor	Apple T2 Chip
MacBook Pro Model: A2251 Reference: MacBook Pro16,2	2020	Intel Core i5-1037NG7 (Ice Lake)	
MacBook Pro Model: A2251 Reference: MacBook Pro16,2	2020	Intel Core i7-1068NG7 (Ice Lake)	
iMac Model: A2115 Reference: iMac20,1	2020	Intel Core i5-10600 (Ice Lake)	
iMac Model: A2115 Reference: iMac20,1	2020	Intel Core i7-10700K (Ice Lake)	
iMac Model: A2115 Reference: iMac20,1	2020	Intel Core i9-10910 (Coffee Lake)	
iMac Model: A2115 Reference: iMac20,2	2020	Intel Core i7-10700K (Ice Lake)	
iMac Model: A2115 Reference: iMac20,2	2020	Intel Core i9-10910 (Coffee Lake)	

Table 3: Platform specifications

Note: The Apple T2 Security Chip is the same exact chip across all platforms. All processing for Cryptography related to FileVault (FDE) is all performed using the Apple T2 / SEP rather than the Intel chipset, so multiple Intel Chips or microarchitectures play no role in the processing (encryption/decryption) and the management of those keys for data under FileVault.

1.3.2 Physical Boundaries

The TOE is comprised of both hardware and software running on the listed platforms as indicated in Table 3. The Encryption Engine (EE) is instantiated on the T2 chip. The AA is instantiated on both the Intel chip (Password Acquisition) and the T2. It contains the Secure Enclave coprocessor which provides security related functionality for all the EE functionality (i.e., other than encryption/decryption of storage data) and all of the cryptographic functionality for AA (i.e., PBKDF2). The Password Acquisition component (AA) is the pre-boot component on the disk and captures the user password and passes it to the T2/SEP. The T2 provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between the storage and main memory of the host platform.

1.3.3 Logical Scope of the TOE

The TOE implements the following security functional requirements from [FDE EE v2.0e] and [FDE AA v2.0e] as listed below:

Cryptographic Support (FCS)

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below:

Algorithms	Standards	CAVP certificates

AES	AES-CBC (as defined in NIST SP 800-38A)	A498(c_asm)
		A497(c_ltc)
		A499(c_glad)
		A494(asm_arm)
		C312, C313, C314, C315, C317, C318, C319, C320, C322,C325, C326, C330, C358,

	AES-GCM (as defined in NIST SP 800-38D)	A498(c_asm)
		A497(c_ltc)

AES	AES-KW (AES as specified in ISO/IEC 18033-3, [NIST SP 800-38F])	A498(c_asm)
		A497(c_ltc)

AES	AES-XTS (AES as specified in ISO/IEC18033-3 and XTS as specified in IEEE 1619)	A494(asm_arm), A497(c_ltc), A498(c_asm)
RSA	FIPS PUB 186-4 Digital Signature Standard (DSS), Appendix B.3.	A495(vng_ltc)

SHS	NIST FIPS Pub 180-4.	A497(c_ltc)
		A495(vng_ltc)
		A500 (vng_neon)

DRBG	CTR_DRBG (AES)	2014, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2028, 2029, C323, C324, C331

Table 4: CAVP References

User Data Protection (FDP)
The TOE encrypts all user data using XTS-AES-128 using a 256-bit key.

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 Disk Utility program operating on macOS invokes management functionality of the AA component in the T2 chip.

Protection of the TSF (FPT)
The TOE implements the following protection of TSF data:

- Protection of Key and Key Material
- Power Saving States
- Timing of Power Saving States
- TSF Testing
- Trusted updates using digital signatures.

The macOS (Operational Environment) retrieves the update package from the Apple update server and forwards the package to the AA component in the T2 chip. The TOE validates the digital signature for the package before it is installed.

1.4 TOE Documentation

The following documents are available in PDF formats.

Documentation	File Format	Date
Apple FileVault 2 on T2 systems running macOS Catalina 10.15 Common Criteria Configuration Guide v0.8	PDF	April 19, 2021
Apple FileVault 2 on T2 systems running macOS Catalina 10.15 Security Target v2.5	PDF	April 19, 2021

Table 5: TOE Documentation

1.5 Other References

- collaborative Protection Profile for Full Drive Encryption – Encryption Engine Version 2.0 + Errata 20190201 [FDE EE v2.0e].
- collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition Version 2.0 + Errata 20190201 [FDE AA v2.0e].

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended and CC Part 3 conformant.

2.2 Protection Profile Conformance

This TOE is conformant to:

- collaborative Protection Profile for Full Drive Encryption – Encryption Engine Version 2.0 + Errata 20190201 [FDE EE v2.0e]. PP Date: February 1, 2019.
- collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition Version 2.0 + Errata 20190201 [FDE AA v2.0e]. PP Date: February 1, 2019.

2.3 Conformance Rationale

This Security Target provides exact conformance to [FDE EE v2.0e] and [FDE AA v2.0e]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to FDE EE v2.0e and FDE AA v2.0e have been addressed. The following table identifies all applicable TDs:

Identifier	Applicable	Exclusion Rationale (if applicable)
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	

Table 6: NIAP Technical Decisions for [FDE EE v2.0e] and [FDE AA v2.0e]

3 Security Problem Definition

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

3.1 Threats

The following threats are drawn directly from the [FDE EE v2.0e] and [FDE AA v2.0e]:

ID	Threat
T.UNAUTHORIZED_DATA_ACCESS	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	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	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	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

ID	Threat
	BEV or otherwise put it in a state in which it discloses protected data to unauthorized users.
T.AUTHORIZATION_GUESSING/EE	Threat agents may exercise host software to repeatedly guess authorization factors, such as passwords and PINs. Successful guessing of the authorization factors may cause the TOE to release DEKs or otherwise put it in a state in which it discloses protected data to unauthorized users.
T.KEYSPACE_EXHAUST	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/EE	Threat agents know plaintext in regions of storage devices, especially in uninitialized regions (all zeroes) as well as regions that contain well known software such as operating systems. A poor choice of encryption algorithms, encryption modes, and initialization vectors along with known plaintext could allow an attacker to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.
T.CHOSEN_PLAINTEXT/EE	Threat agents may trick authorized users into storing chosen plaintext on the encrypted storage device in the form of an image, document, or some other file. A poor choice of encryption algorithms, encryption modes, and initialization vectors along with the chosen plaintext could allow attackers to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.
T.UNAUTHORIZED_UPDATE	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_FIRMWARE_MODIFY/EE	An attacker attempts to modify the firmware in the SED via a command from the AA or from the

ID	Threat
	host platform that may compromise the security features of the TOE.
T.UNAUTHORIZED_FIRMWARE_MODIFY	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.

Table 7: Threats

3.2 Assumptions

The following assumptions are drawn directly from the [FDE EE v2.0e] and [FDE AA v2.0e]:

ID	Assumption
A.INITIAL_DRIVE_STATE	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.SECURE_STATE	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	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

ID	Assumption
	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	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	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	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	External tokens that contain authorization factors are used for no other purpose than to store the external token authorization factors.
A.POWER_DOWN	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.PASSWORD_STRENGTH	Authorized administrators ensure password/passphrase authorization factors have sufficient strength and entropy to reflect the sensitivity of the data being protected.

ID	Assumption
A.PLATFORM_I&A	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	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	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.

Table 8: Assumptions

3.3 Organizational Security Policies

The [FDE EE v2.0e] and [FDE AA v2.0e] do not define any OSPs.

4 Security Objectives

The security objectives for the TOE have been taken from [FDE EE v2.0e] and [FDE AA v2.0e] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operational Environment
OE.TRUSTED_CHANNEL	Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure.
OE.INITIAL_DRIVE_STATE	The OE provides a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption.
OE.PASSPHRASE_STRENGTH	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	Volatile memory is cleared after power-off so memory remnant attacks are infeasible.
OE.POWER_DOWN/EE	Volatile memory is cleared after entering a Compliant power saving state or turned off so memory remnant attacks are infeasible.
OE.SINGLE_USE_ET	External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.
OE.TRAINED_USERS	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.
OE.STRONG_ENVIRONMENT_CRYPTO	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	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.
OE.PHYSICAL	The Operational Environment will provide a secure physical computing space such that an adversary is not able to make modifications to the environment or to the TOE itself.
OE.PLATFORM_STATE	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	The Operational Environment will provide individual user identification and authentication mechanisms

ID	Objective for the Operational Environment
	that operate independently of the authorization factors used by the TOE.

Table 9: Objectives for the Operational Environment

5 Extended Security Functional Components

Requirements	Descriptions
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_KYC_EXT.1	Key Chaining (Initiator)
FCS_KYC_EXT.2	Key Chaining (Recipient)
FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning
FCS_RBG_EXT.1	Random Bit Generation
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FCS_VAL_EXT.1	Validation
FDP_DSK_EXT.1	Protection of Data on Disk
FPT_FAC_EXT.1	Firmware Access Control
FPT_FUA_EXT.1	Firmware Update Authentication
FPT_KYP_EXT.1	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States
FPT_TUD_EXT.1	Trusted Update
FPT_TST_EXT.1	TSF Testing

Table 10: Extended Security Functional Components

5.1 Extended Security Functional Components Rationale

The definition of all SFRs with the appendix of "_EXT" is supplied by the protection profile. All extended security functional components are derived directly from [FDE EE v2.0e] and [FDE AA v2.0e] and applied verbatim.

6 Security Requirements

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017 and all international interpretations.

Requirements	Descriptions
FCS_AFA_EXT.1	Authorization Factor Acquisition
FCS_AFA_EXT.2	Timing of Authorization Factor Acquisition
FCS_CKM.1(a)	Cryptographic Key Generation (Asymmetric Keys)
FCS_CKM.1(b)	Cryptographic key generation (Symmetric Keys)
FCS_CKM.1(c)	Cryptographic Key Generation (Data Encryption Key)
FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
FCS_CKM.4(b)	Cryptographic Key Destruction (TOE-Controlled Hardware)
FCS_CKM.4(d)	Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
FCS_CKM_EXT.6	Cryptographic Key Destruction Types
FCS_COP.1(a)	Cryptographic Operation (Signature Verification)
FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
FCS_COP.1(c)	Cryptographic Operation (Message Authentication)
FCS_COP.1(d)	Cryptographic operation (Key Wrapping)
FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1(g)	Cryptographic Operation (Key Encryption)
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_KYC_EXT.1	Key Chaining (Initiator)
FCS_KYC_EXT.2	Key Chaining (Recipient)
FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning
FCS_RBG_EXT.1	Random Bit Generation
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FCS_VAL_EXT.1	Validation
FDP_DSK_EXT.1	Protection of Data on Disk
FMT_MOF.1	Management of Functions Behavior
FMT_SMF.1(1)	Specification of Management Functions - Authorization Acquisition
FMT_SMF.1(2)	Specification of Management Functions - Encryption Engine
FMT_SMR.1	Security Roles
FPT_FAC_EXT.1	Firmware Access Control
FPT_FUA_EXT.1	Firmware Update Authentication
FPT_KYP_EXT.1(1)	Protection of Key and Key Material
FPT_KYP_EXT.1(2)	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States

Requirements	Descriptions
FPT_TUD_EXT.1	Trusted Update
FPT_TST_EXT.1	TSF Testing

Table 11: SFRs

6.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3);
- Where operations were completed in the PP or EP itself, the formatting used in the PP or EP has been retained.

Extended SFRs are identified by having a label 'EXT' after the requirement name. Formatting conventions outside of operations matches the formatting specified within the PP or EP.

6.2 Security Functional Requirements

6.2.1 Cryptographic Support (FCS)

6.2.1.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

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.

].

6.2.1.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

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

6.2.1.3 FCS_CKM.1(a) Cryptographic Key Generation (Asymmetric Keys)

FCS_CKM.1.1(a) Refinement The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***RSA schemes using cryptographic key sizes of [2048-bit] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)",Appendix B.3;***

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

]

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

FCS_CKM.1.1(b) Refinement 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*].

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

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

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

and specified cryptographic key sizes [***256 bits***] ~~that meet the following: [*assignment: list of standards*].~~

6.2.1.6 FCS_CKM.4(a) Cryptographic Key Destruction (Power Management) - Authorization Acquisition

FCS_CKM.4.1(a) Refinement The TSF shall [**erase**] **cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1** that meets the following: [*a key destruction method specified in FCS_CKM_4(d)*].

6.2.1.7 FCS_CKM.4(a) Cryptographic Key Destruction (Power Management) - Encryption Engine

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

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

FCS_CKM.4.1(b) Refinement 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*** [
  - o ***single overwrite consisting of [***
    - ▪ ***zeroes,***

    ]
  - o ***removal of power to memory,***

  ];
- ***For non-volatile memory [***
  - o ***that employs a wear-leveling algorithm, the destruction shall be executed by a [***
    - ▪ ***Single overwrite consisting of zeroes***

];

]

]

that meets the following: [*no standard*].

6.2.1.9 FCS_CKM.4(d) Cryptographic Key Destruction (Software TOE, 3rd Party Storage) - Authorization Acquisition

FCS_CKM.4.1(d) Refinement: 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 [

  - o ***single overwrite consisting of [***
    - ▪ ***zeroes,***

  ]

  - o ***removal of power to memory,***

  ];

]

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

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

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

FCS_CKM_EXT.4.1(b) Refinement: 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.**

6.2.1.12 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

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

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

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

- ***RSA Digital Signature Algorithm with a key size (modulus) of [2048-bit];***

]

that meets 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***

].

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

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

6.2.1.15 FCS_COP.1(c) Cryptographic Operation (Message Authentication)

FCS_COP.1.1(c) 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”*].

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

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***]].

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

FCS_COP.1.1(f) The TSF shall perform [*data encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES used in* [***XTS*** mode] and cryptographic key sizes [***256 bits***] that meet the following: [*AES as specified in ISO/IEC18033-3*, [***XTS as specified in IEEE 1619***]].

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

FCS_COP.1.1(g) The TSF shall perform [*key encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES used in* [***CBC, GCM***] 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, GCM as specified in ISO/IEC 19772***]].

6.2.1.19 FCS_KDF_EXT.1 Cryptographic Key Derivation

FCS_KDF_EXT.1.1 The TSF shall accept [a conditioned password submask] to derive an intermediate key, as defined in [

- NIST SP 800-132],

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

6.2.1.20 FCS_KYC_EXT.1 Key Chaining (Initiator)

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

- intermediate keys originating from one or more submask(s) to the BEV using the following method(s):
  - o [key derivation as specified in FCS_KDF_EXT.1]

]

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.

].

6.2.1.21 FCS_KYC_EXT.2 Key Chaining (Recipient)

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.

6.2.1.22 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

FCS_PCC_EXT.1.1 A password used by the TSF to generate a password authorization factor shall enable up to [*256*] characters in the set of {upper case characters, lower case characters, numbers, and [*all other 8-bit values*]} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[SHA-256], with [*50,000*] iterations, and output cryptographic key sizes [256 bits] that meet the following: [*NIST SP 800-132*].

6.2.1.23 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

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

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

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

]

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

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

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

FCS_SNI_EXT.1.2 The TSF shall use [unique nonces with a minimum size of [*64*] bits].

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

- CBC: IVs shall be non-repeating and unpredictable;
- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer;
- GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key].

6.2.1.25 FCS_VAL_EXT.1 Validation

FCS_VAL_EXT.1.1 The TSF shall validate a BEV using the following method(s): [

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

].

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

FCS_VAL_EXT.1.3 The TSF shall [

- block validation after [*10*] of consecutive failed validation attempts,

].

6.2.2 Class FDP: User Data Protection

6.2.2.1 FDP_DSK_EXT.1 Extended: Protection of Data on Disk

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.

6.2.3 Class FMT: Security Management

6.2.3.1 FMT_MOF.1 Management of Functions Behavior

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

6.2.3.2 FMT_SMF.1(1) Specification of Management Functions - Authorization Acquisition

FMT_SMF.1.1(1) Refinement: 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 factors or set of authorization factors used.*

d) *initiate TOE firmware/software updates,*

e) [***configure authorization factors***]

6.2.3.3 FMT_SMF.1(2) Specification of Management Functions - Encryption Engine

FMT_SMF.1.1(2) The TSF shall be capable of performing the following management

functions:

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

b) *erase the DEK, as specified in FCS_CKM.4(a)*

c) *initiate TOE firmware/software updates,*

d) [*no other functions*]

6.2.3.4 FMT_SMR.1 Security Roles - Authorization Acquisition

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

FMT_SMR1.2 The TSF shall be able to associate users with roles.

6.2.4 Class FPT: Protection of the TSF

6.2.4.1 FPT_FAC_EXT.1 Firmware Access Control

FPT_FAC_EXT.1.1 The TSF shall require [a password] before the firmware update proceeds.

6.2.4.2 FPT_FUA_EXT.1 Firmware Update Authentication

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.

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

6.2.4.3 FPT_KYP_EXT.1(1) Extended: Protection of Key and Key Material (AA)

FPT_KYP_EXT.1.1(1) The TSF shall [

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

].

6.2.4.4 FPT_KYP_EXT.1(2) Extended: Protection of Key and Key Material (EE)

FPT_KYP_EXT.1.1(2) 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 of the following criteria [
  - The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.2]

]

6.2.4.5 FPT_PWR_EXT.1 Power Saving States

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

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

6.2.4.6 FPT_PWR_EXT.2 Timing of Power Saving States

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

6.2.4.7 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

The TSF shall verify updates to the TOE software, firmware using a [digital signature] by the manufacturer prior to installing those updates.

6.2.4.8 FPT_TST_EXT.1 Extended: TSF Testing

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: [

- **authenticity and integrity check of software/firmware**
- **Known Answer Tests (KATs)**
    - **CTR_DRBG with AES 256**
    - **RSA 2048 with SHA-256 signature verification**
    - **RSA 2048 with SHA-256 encryption/decryption**
    - **HMAC-SHA-256 MAC generation**
    - **AES 128 XTS encrypt and decrypt**
    - **AES CBC 256-bit encrypt and decrypt**
    - **AES GCM 256-bit encrypt and decrypt**

].

6.3 TOE SFR Dependencies Rationale for SFRs

[FDE EE v2.0e] and [FDE AA v2.0e] contain all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the cPPs have been approved.

6.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [FDE EE v2.0e] and [FDE AA v2.0e] which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development (ADV)	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance

Assurance Class	Components	Components Description
(AGD)	AGD_PRE.1	Preparative Procedures
Life cycle Support (ALC)	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests (ATE)	ATE_IND.1	Independent Testing – Sample
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability Survey

Table 12: Security Assurance Requirements

6.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

6.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Apple to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this cPP will may interfaces to the Operational Environment that are not directly invoked by TOE users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" documentation is necessary to satisfy the Evaluation Activities specified in the SD. The Evaluation Activities in the SD are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.
AGD_OPE.1	The operational user guidance does not have to be contained in a single document. Guidance to users, administrators, and integrators can be spread among documents or web pages.

SAR Component	How the SAR will be met
	The developer should review the Evaluation Activities contained in the SD to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.
AGD_PRE.1	As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.
ALC_CMC.1	This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user. The evaluator performs the CEM work units associated with ALC_CMC.1
ALC_CMS.1	Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMS.1.
ATE_IND.1	Apple will provide the TOE for testing.
AVA_VAN.1	Apple will provide a document identifying the list of software and hardware components.

Table 13: TOE Security Assurance Measures

7 TOE Summary Specification

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

TOE SFRs	Rationale
FCS_AFA_EXT.1/FCS_PCC_EXT.1	The TOE supports password authentication factor. Passwords of up to 256 characters are supported and can be comprised of any combination of upper-case characters, lower case characters, numbers, and any other 8-bit special character. The TOE supports a password authorization factor. For password-based authentication, the user’s password, the TOE’s UID and a salt value are used to perform a password-based derivation function (PBKDF2) and derive the Unlock Key. The UID is prefixed to the Salt value. The Unlock Key is defined as the Border Encryption Value (BEV) and is used to unwrap the Class Key with the AES Key Wrap (KW) algorithm. The password is validated if the AES KW function does not return a “Fail” result. The Key derivation function is implemented according to NIST SP 800-132. It leverages the HMAC-SHA-256 algorithm with 50,000 iterations and the UID as the “purpose” value as defined in Appendix A.2.1 of SP 800-132.
FCS_AFA_EXT.2	To resume from a compliant power state, one must re-authenticate to the TOE. The user can authenticate using username and password.
FCS_CKM.1(a)	The TOE supports RSA schemes using cryptographic key sizes of 2048-bit which meets FIPS PUB 186-4, “Digital Signature Standard (DSS)”,Appendix B.3;
FCS_CKM.1(b)	The TOE generates symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 with 256 bits key size. Refer Table 4: CAVP References in this

TOE SFRs	Rationale
	document for algorithm testing certificates.
FCS_CKM.1(c)	All symmetric and asymmetric cryptographic keys are randomly generated internal to the TOE using the SEP’s True Random Number Generator (TRNG). The SEP’s TRNG is seeded by 24 ring oscillators and post processed with an SP 800-90A CTR_DRBG. The ring oscillators are constantly inputting new noise data into the conditioner (SHA-256 hash) from which the DRBG seed is obtained. Thus, the conditioner accumulates the entropy of the ring oscillators. 0.9 bits of entropy is provided per data bit. Full entropy of 256 bits is reached after collecting 285 bits of data from the noise source. As the noise source runs faster than the DRBG, the number of data bits collected from the noise source and injected into the conditioner is always considered higher than 285 bits. Thus, the DRBG is seeded with greater than 256 bits of entropy. Key generation using the DRBG are performed by calling the DRBG’s generate function. The Volume Key is defined as the Data Encryption Key (DEK). It is randomly generated when a user volume is created, and the key is destroyed by issuing an authenticated command by a single overwrite consisting of zeroes. On Mac computers with the T2 chip, all FileVault key handling occurs in the Secure Enclave; encryption keys are never directly exposed to the Intel CPU. All APFS volumes are created with a volume key by default. Volume and metadata contents are encrypted with this volume key, which is wrapped with the class key. The class key is protected by a combination of the user’s password and the hardware UID when FileVault is turned on. This

TOE SFRs	Rationale
	protection is the default on Mac computers with the T2 chip. Note: Encryption of removable storage devices does not utilize the security capabilities of the Apple T2 Security Chip, and its encryption is performed in the same manner as Mac computers without the T2 chip. • The VEK is generated in the SEP • The VEK is wrapped by a KEK • The KEK is derived from multiple pieces (UID, User Passcode (PBKDF2), Tangling) —> Unlock Key • Keys and wrapping of Keys are destroyed within the SEP. When deleting a volume, its volume key is securely deleted by Secure Enclave. This prevents future access with this key even by the Secure Enclave. Refer Table 4: CAVP References in this document for algorithm testing certificates.
FCS_CKM.4(a)/FCS_CKM.4(b)/FCS_CKM.4(d)/ FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b) and FCS_CKM_EXT.6 and FPT_KYP_EXT.1(1) and FPT_KYP_EXT.1(2)	The TOE leverages NAND flash for non-volatile memory. All symmetric keys that are persistently stored, except for the UID, are wrapped in NAND flash. The UID is fused into the SEP's ROM is not accessible by any component outside of the SEP and cannot be erased. The TOE erases cryptographic keys and key material from volatile memory by performing a single overwrite of zeroes and/or by removal of power to the memory. The TOE erases cryptographic keys and key material from non-volatile memory by performing a single overwrite of zeroes. The TOE leverages DRAM for volatile memory. Keys are stored in volatile memory while being used for their specific operation. Except for the UID and the Unlock Key, all symmetric keys are

TOE SFRs	Rationale
	introduced into volatile memory after being randomly generated or by unwrapping or decrypting a key stored in non-volatile memory. The Unlock Key is introduced into volatile memory after the password-based derivation process has been completed. The TOE will destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1. Keys are only stored in volatile memory when they are required to perform a specific cryptographic operation. Since the keys are being used by the SEP to perform the operation, the SEP tracks the memory location of the key until the operation is complete. Once the keys are no longer required, the key that was used to perform the specific operation is erased from volatile memory by performing a single overwrite of zeroes. The erase operation is performed by the SEP and is not configurable by a user. There are no circumstances that do not conform to the key destruction requirement (e.g. sudden unexpected power loss). The SEP performs the wrapping of keys, which are then sent to the memory controller for storage. The memory controller takes the block of data and the memory location provided by the SEP and stores the data in memory.
FCS_COP.1(a)	Signature verification is done as part of the Secure Boot process, for firmware and software updates. Signatures are verified using RSA 2048-bit and SHA-256. The CA Public Key is embedded in the SEP’s Boot ROM code in manufacturing and is used for all macOS running on Mac hardware with Apple T2 chip. The TOE image is signed using this key’s corresponding private key.

TOE SFRs	Rationale
FCS_COP.1(b)	The TOE supports SHA-256 algorithm to perform digital signature verification of 2048 bit RSA keys and in HMAC operations.
FCS_COP.1(c)	The TOE supports keyed hash algorithm with HMAC-SHA-256 supporting key size of 256 bits and block size of 512 bits. SHA-256 hashing function is used.
FCS_COP.1(d)	When the User requests a crypto service from the module, it must provide the passcode and a reference to the user keybag that is stored encrypted under SP800-38F AES Key Wrapping (AES-KW) within SKS. The module uses PBKDF2 to derive an AES key from the Operator provided passcode. The derived AES key is then used by the module's SP800-38F AES Key Unwrapping function (i.e. AES-KW-AD3) to decrypt the referenced user keybag and to verify the authenticity of the decrypted key. As AES-KW is an authentication cipher, the decryption operation will only succeed without an authentication error. This implies that the user provided the correct passcode to derive the correct AES key for AES Key Unwrapping. Any other passcode will derive a different AES key which will result in a wrong decrypted user key that fails the authentication check. If the user keybag can be successfully unwrapped, the user is authenticated to the module and the requested crypto service will then be proceeded with the unwrapped user key. The failure of unwrapping user keybag is also a user authentication failure and the Operator will be denied access to the module.
FCS_COP.1(f)	The TOE supports AES data encryption and AES decryption using AES-128 in XTS mode. The key size supported is 256 bits.
FCS_COP.1(g)	The TOE supports key encryption and decryption using AES algorithm. The modes supported are CBC and GCM modes. The key size supported is 256 bits.

TOE SFRs	Rationale
FCS_KDF_EXT.1	The Key derivation function is implemented according to NIST SP 800-132. It leverages the HMAC-SHA-256 algorithm with 50,000 iterations and the UID as the "purpose" value as defined in Appendix A.2.1 of SP 800-132. The Unlock Key is defined as the Boarder Encryption Value (BEV) and is used to unwrap the Class Key with the AES Key Wrap (KW) algorithm.
FCS_KYC_EXT.1 and FCS_KYC_EXT.2	The TOE supports BEV sizes of 256 bits. The TOE maintains a 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).
FCS_RBG_EXT.1	The TOE performs deterministic random bit generation services according to NIST SP 800-90A using CTR_DRBG (AES). The SEP TRNG is seeded by 24 ring oscillators. The ring oscillators are constantly inputting new noise data into the conditioner (SHA-256 hash) from which the DRBG seed is obtained. The full entropy of 256 bits is achieved after collecting 285 bits of data from the noise source.
FCS_SNI_EXT.1	The TOE can generate salts, nonces, and initialization vectors (IVs) using the SEP's DRBG. The DRBG is seeded by the SEP's hardware TRNG. Salts are 16 bytes and are used with the PBKDF2. Nonces are 8 bytes and are used with the trusted update process. The IV used with the AES CBC and AES CBC is non-repeating and unpredictable. The TOE enforces that number. The number of invocations of GCM does not exceed 2^32 for a given secret key. Tweaks are used with the AES XTS mode of operation. The tweak values should be non-

TOE SFRs	Rationale
	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	The TOE will validate a BEV using key wrap as specified in FCS_COP.1(d). The TOE requires the validation of the BEV prior to allowing access to TSF data after exiting a Compliant power saving state and it will block validation after 10 consecutive failed validation attempts.
FDP_DSK_EXT.1	The T2 provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between the flash storage and the main memory of the host platform. The T2 chip is placed in the middle of the data path between the Intel chip and the storage disk. The T2 performs the encryption/ decryption of the data prior to reaching the Intel chip or the storage. When a read operation is made, the data must first be decrypted by the T2 before the Intel chip has access to the data. When a write operation is made, the data is first encrypted by the T2 and then written to memory as a block of encrypted data. This arrangement ensures that standard methods of accessing the disk 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 2) 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 connected to another host platform.

TOE SFRs	Rationale
	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, 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 2 when provisioning the host platform at first run, FileVault 2 can be enabled later through the Security & Privacy menu available via the host platform. By default, the host platform's storage drive is always encrypted. The TOE cryptographic key management changes after enabling FileVault 2. 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 T2. If FileVault is disabled and re-enabled, a new recovery key is generated.
FMT_MOF.1	The TOE restricts the ability to modify the behavior of complaint power saving state to authorized users.
FMT_SMR.1	The TOE supports authorized user role and it can associate users to roles.

FMT_SMF.1(1)
FMT_SMF.1(2)
	The TOE supports the following management functions:
	• Authorization Acquisition:
	• Forwarding requests to change the DEK to the EE,
	• The Volume Key is defined as the Data Encryption Key (DEK). It is randomly generated when a user volume is

TOE SFRs	Rationale
	created, and the key is destroyed by issuing an authenticated command by a single overwrite consisting of zeroes. • The DEK is the Volume key which is created for each volume at volume creation time. • The user can destroy the Volume key by destroying/erasing the volume. This option can be selected after authenticating to the TOE and the TOE performs a cryptographic erase of the keying material. • The above can be achieved by starting the Disk Utility application and then selecting the appropriate volume to be erased. • Forwarding requests to cryptographically erase the DEK to the EE • The Volume Key is defined as the Data Encryption Key (DEK). It is randomly generated when a user volume is created, and the key is destroyed by issuing an authenticated command by a single overwrite consisting of zeroes. • The DEK is the Volume key which is created for each volume at volume creation time. • The user can destroy the Volume key by destroying/erasing the volume. This option can be selected after authenticating to the TOE and the TOE performs a cryptographic erase of the keying material. • The above can be achieved by starting the Disk Utility application and then selecting the appropriate volume to be erased. • allowing authorized users to change authorization factors or set of authorization factors used • Once the user successfully authenticates to the TOE, the TOE can be configured to change the

TOE SFRs	Rationale
	authorization factors that can be used: password. • The above can be achieved by navigating to System Preferences-> 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 Preferences-> Users & Groups -> Select the appropriate user -> Change Password. • Authorization Acquisition and Encryption Engine • Forwarding requests to change the DEK to the Encryption Engine. • Forwarding requests to cryptographically erase the DEK to the Encryption Engine. • initiate TOE firmware/software updates • The user must successfully login to the TOE before initiating a TOE firmware/software update. After successfully authenticating to the TOE, the user manually downloads the TOE software update(s) from https://support.apple.com/downloads. • Once the update(s) is downloaded, the user needs to initiate the TOE update process by double clicking or right-click -> Open the downloaded update. • configure cryptographic functionality • The Volume Key is defined as the Data Encryption Key (DEK). It is randomly generated when a user volume is created, and the key is destroyed by issuing an authenticated command by a single overwrite consisting of zeroes.

TOE SFRs	Rationale
	• The DEK is the Volume key which is created for each volume at volume creation time. • The user can destroy the Volume key by destroying/erasing the volume. This option can be selected after authenticating to the TOE and the TOE performs a cryptographic erase of the keying material. • The above can be achieved by starting the Disk Utility application and then selecting the appropriate volume to be erased.
FPT_FAC_EXT.1/ FPT_FUA_EXT.1/FPT_TUD_EXT.1	The user must successfully login to the TOE before initiating a TOE software/firmware update. Only authorized users i.e. privileged/non-privileged users can initiate the TOE update process. A vendor-controlled server is leveraged for obtaining firmware update code packages. The code packages containing the macOS, T2 OS/firmware, and SEP OS/firmware are all bundled together. The firmware/OS is stored within the T2 chip. The TOE stores the incoming update in a temporary location on flash. Once the transfer is complete, the SEP verifies the RSA 2048-bit digital signature verification. If the verification is successful, the TOE installs the update and reboots the host device. If the verification is unsuccessful, the TOE terminates the updates process. The Mac operating system and software application updates can be downloaded manually through the following website: https://support.apple.com/downloads
FPT_PWR_EXT.1/FPT_PWR_EXT.2	The TOE supports the following power savings state: G2(S5)-soft off. The TOE can enter G2(S5)-soft off power savings state by the user selecting Shutdown option on the TOE host device.
FPT_TST_EXT.1	During power-up, the TOE performs a signature verification of firmware and software using the Apple Root CA Public

TOE SFRs	Rationale
	Key. When the host device is powered-on, the SEP initiates the Secure Boot process. The SEP's Boot ROM first authenticates the signature of the Bridge Boot code (T2 Boot ROM code). If the verifications fails, the TOE returns an error and enters the Device Firmware Upgrade (DFU) mode; requiring a correct update to continue. If the verification is successful, the Bridge Boot code then authenticates the signature of the T2 kernel cache. The T2 kernel cache then authenticates the signature of the Unified Extensible Firmware Interface (UEFI) firmware. The UEFI firmware is then used to authenticate the boot.efi file within the Intel processor of the TOE host device. The boot.efi file then authenticates the macOS immutable kernel. The macOS then authenticates third party kernel extensions (kexts) and OS Userspace. The TOE performs the following known answer tests (KATs) to verify the correct operation of the cryptographic functions: • CTR_DRBG with ASE 256: The TOE instantiates the DRBG with a known value, invokes the generate function, and compares the generated bits to the expected bits. • RSA 2048 with SHA-256 Signature Verification: satisfied by the Firmware Integrity signature verification test above. • RSA 2048 with SHA-256 Encrypt/Decrypt • HMAC-SHA-256: MAC generation with a known key and message. • AES 128 XTS Encrypt/Decrypt: This shows the correct operation of AES Encrypt and Decrypt primitive functions with a 256-bit key. • AES CBC 256-bit encrypt and decrypt KATs • AES GCM 256-bit encrypt and decrypt KATs

Table 14: TOE Summary Specification SFR Description

8 Annex A: References

Identifiers	Descriptions
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-004
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-56A]	NIST Special Publication 800-56A Rev 2, May 2013
[800-56B]	NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009
[800-38A]	[NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-38D]	NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.
[800-38F]	NIST Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping.

Table 15: Annex A References