Assurance Activity Report

Apple macOS 26 Tahoe: FileVault

VID 11696

Version 1.1

Evaluated by:

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

United States of America

Validation Body:

NIAP-CCEVS

TOE Developer and Sponsor:

Apple Inc.

Report Authorized by:

Trang Huynh

This report must not be used to claim product certification, approval, or endorsement by NIAP CCEVS, NVLAP, NIST, or any agency of the Federal Government.

Classification Note

Public Information (public)

This classification level is for information that may be made available to the general public. No specific security procedures are required to protect the confidentiality of this information. Information classified “public” may be freely distributed to anyone inside or outside of atsec.

Information with this classification shall be clearly marked “public”, except that it is not required to mark “public” on printed marketing material obviously intended for publication.

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2026-06-29	Joachim Vandersmissen	First version
1.1	2026-07-28	Joachim Vandersmissen	Address validator feedback

Table of Contents

List of Tables

1 Evaluation Basis and Documents

This evaluation is based on the “Common Criteria for Information Technology Security Evaluation” Version 3.1 Revision 5 [CC], the “Common Methodology for Information Technology Security Evaluation” [CEM], and the additional assurance activities defined in the following:

- [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, dated 2024-05-31
    - [CPP_FDE_AA_V2.0E]: collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0 + Errata 20190201, dated 2019-02-01
    - [CPP_FDE_EE_V2.0E]: collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0 + Errata 20190201, dated 2019-02-01

This evaluation claims exact compliance with the above PP-Configuration and PPs.

The following scheme documents and interpretations have been considered:

- [CCEVS-LG]: "CCEVS Labgrams", version as of July 2026.
- [CCEVS-PL]: "CCEVS Scheme Policy Letters", version as of July 2026.
- [CCEVS-PUB]: "CCEVS Scheme Publications", version as of July 2026.
- [CCEVS-TD]: "Technical Decisions", version as of July 2026.

2 Evaluation Results

The evaluator work units have been performed, including: evaluator actions and analysis explicitly stated in the CEM; evaluator actions implicitly derived from developer action elements described in the CC Part 3; and evaluator confirmation that requirements for content and presentation of evidence elements described in the CC Part 3 have been met.

The evaluation was performed by informal analysis of the evidence provided by the sponsor.

2.1 CAVP Summary

The TOE uses the following cryptographic implementations:

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

The tables below show the cryptographic services used by the TOE and provided by the cryptographic implementations that are included in the TOE, describing cryptographic operations, the algorithm names, and the applicable standard. 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 1: Mapping of SFRs to CAVP certificates (USR)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(a)	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384	[FIPS186-5]	A7643

		Padding: PKCS#1 v1.5		
FCS_COP.1(b)	SHA2-384	Byte-oriented mode	[FIPS180-4]	A7645

Table 2: Mapping of SFRs to CAVP certificates (KRN)

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

Table 3: Mapping of SFRs to CAVP certificates (SKS)

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

Table 4: Mapping of SFRs to CAVP certificates (SE)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(g)	AES-CBC	256 bits encrypt	[SP800-38A]	A3496, A6547
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A3490, A6548, A6924

Table 5: Mapping of SFRs to CAVP certificates (DMA)

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

2.2 Security Functional Requirements

2.2.1 Cryptographic Support (FCS)

2.2.1.1 Authorization Factor Acquisition (FCS_AFA_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.1-ASE-01

The evaluator shall first examine the TSS to ensure that the authorization factors specified in the ST are described. For password-based factors the examination of the TSS section is performed as part of FCS_PCC_EXT.1 Evaluation Activities. Additionally in this case, the evaluator shall verify that the operational guidance discusses the characteristics of external authorization factors (e.g., how the authorization factor must be generated; format(s) or standards that the authorization factor must meet) that are able to be used by the TOE.

If other authorization factors are specified, then for each factor, the TSS specifies how the factors are input into the TOE.

Summary

Section 7.1.1.1 of the [ST] specifies that the TOE only accepts password authorization factors. The examination of this section is performed as part of FCS_PCC_EXT.1 Evaluation Activities. The TOE does not support external authorization factors.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.1-AGD-01

The evaluator shall verify that the AGD guidance includes instructions for all of the authorization factors. The AGD will discuss the characteristics of external authorization factors (e.g., how the authorization factor is generated; format(s) or standards that the authorization factor must meet, configuration of the TPM device used) that are able to be used by the TOE.

Summary

Sections 3.3.1 and 3.3.3 of the [CCGUIDE] provide instructions for setting and changing the password. The TOE does not support external authorization factors.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-AFA_EXT.1-AKM-01

The evaluator shall examine the Key Management Description to confirm that the initial authorization factors (submasks) directly contribute to the unwrapping of the BEV.

The evaluator shall verify the KMD describes how a submask is produced from the authorization factor (including any associated standards to which this process might conform), and verification is performed to ensure the length of the submask meets the required size (as specified in this requirement).

Summary

Section 2.1 of the [KMD] explains that the password authentication factor is used to derive the Password-Derived Key (PDK), which serves as the submask and BEV. This key is 256 bits long, derived using PBKDF2 and AES encryption (one round of AES followed by repeated AES-CBC), and the derivation process is compliant with SP 800-132.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.1-ATE-01

The password authorization factor is tested in FCS_PCC_EXT.1.

The evaluator shall also perform the following tests:

Test 1 (conditional): If there is more than one authorization factor, ensure that failure to supply a required authorization factor does not result in access to the decrypted plaintext data.

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE and verified that the keys were not loaded if the password is not entered. Then, the evaluator logged in using the password and verified the keys were available.

As these keys are used to encrypt the plaintext data, the decrypted plaintext data is only accessible if the keys are available. In other words, the correct password must be supplied to access the decrypted plaintext data.

2.2.1.2 Timing of Authorization Factor Acquisition (FCS_AFA_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.2-ASE-01

The evaluator shall examine the TSS for a description of authorization factors and which of the factors are used to gain access to user data after the TOE entered a Compliant power saving state. The TSS is inspected to ensure it describes that each authorization factor satisfies the requirements of FCS_AFA_EXT.1.1.

Summary

Section 7.1.1.2 of the [ST] specifies that the username and password are used to exit a Compliant power saving state. The evaluator verified that the required authorization factors satisfy the requirements of FCS_AFA_EXT.1.1.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.2-AGD-01

The evaluator shall examine the guidance documentation for a description of authorization factors used to access plaintext data when resuming from a Compliant power saving state.

Summary

Section 3.2 of the [CCGUIDE] discusses the lists the authorization factors used to access plaintext data when resuming from a Compliant power saving state.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_AFA_EXT.2-ATE-01

The evaluator shall perform the following test:

- *Enter the TOE into a Compliant power saving state*

- *Force the TOE to resume from a Compliant power saving state*
- *Release an invalid authorization factor and verify that access to decrypted plaintext data is denied*
- *Release a valid authorization factor and verify that access to decrypted plaintext data is granted.*

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE, causing it to enter the Compliant power saving state. Then, the evaluator entered an incorrect password and verified that the keys were not loaded. Finally, the evaluator logged in using the correct password and verified the keys were available.

As these keys are used to encrypt the plaintext data, the decrypted plaintext data is only accessible if the keys are available. In other words, the correct password must be supplied to access the decrypted plaintext data.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.1-B-ASE-01

Assurance Activity AA-FDEEECPP-FCS_CKM.1-B-ASE-01

The evaluator shall review the TSS to determine that a symmetric key is supported by the product, that the TSS includes a description of the protection provided by the product for this key. The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE.

Summary

Section 7.1.1.3 of the [ST] identifies the Volume Encryption Key as a symmetric key supported by the TOE. The evaluator verified that the protection of the key is described (wrapped using a Key Encryption Key), and the key size supported by the TOE is specified.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.1-B-AGD-01

Assurance Activity AA-FDEEECPP-FCS_CKM.1-B-AGD-01

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key size(s) for all uses specified by the AGD documentation and defined in this cPP.

Summary

Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.1-B-AKM-01

Assurance Activity AA-FDEEECPP-FCS_CKM.1-B-AKM-01

If the TOE uses a symmetric key as part of the key chain, the KMD should detail how the symmetric

key is used as part of the key chain.

Summary

Section 2.1 of the [KMD] explains that the TOE uses the System Measurement Device Key (SMDK) and Secure Enclave Unique ID (UID) as part of the process to derive the BEV.

Section 2.2 of the [KMD] explains that the TOE uses four symmetric keys in the keychain from BEV to DEK: the Volume Encryption Key (VEK), the Key Encryption Key (KEK), the Root Encryption Key (REK), and the Media Key. The VEK is wrapped by the KEK and the media key. The KEK is wrapped by the REK, which is obtained by unlocking a counter lockbox with the BEV.

Test Assurance Activities

No assurance activities defined.

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

TSS Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.1-C-ASE-01

The evaluator shall examine the TSS to determine that it describes how the TOE obtains a DEK (either generating the DEK or receiving from the environment).

If the TOE generates a DEK, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. If the DEK is generated outside of the TOE, the evaluator checks to ensure that for each platform identified in the TOE the TSS, it describes the interface used by the TOE to invoke this functionality. The evaluator uses the description of the interface between the RBG and the TOE to determine that it requests a key greater than or equal to the required key sizes.

If the TOE received the DEK from outside the host platform, then the evaluator shall examine the TSS to determine that the DEK is sent wrapped using the appropriate encryption algorithm.

Summary

Section 7.1.1.3 of the [ST] states that the TOE generates the DEK (Volume Encryption Key) using the random bit generator specified in FCS_RBG_EXT.1. The DEK is generated internally when the data volume is created.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.1-C-AKM-01

If the TOE received the DEK from outside the host platform, then the evaluator shall verify that the KMD describes how the TOE unwraps the DEK.

Summary

Section 4.4 of the [KMD] states that the TOE does not import any DEKs. Therefore, this assurance activity is not applicable.

Test Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.1-C-ATE-01

The evaluator shall perform the following tests:

Test 1: The evaluator shall configure the TOE to ensure the functionality of all selections.

Summary

The evaluator used a tool provided by the vendor to read the wrapped data encryption keys loaded by the Secure Enclave. The evaluator verified that these values changed when the TOE was factory reset, implying that new data encryption keys were generated during factory reset, using the TOE's RBG.

2.2.1.5 Cryptographic Key Destruction (Power Management) (FCS_CKM.4(a))

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-A-ASE-01

Assurance Activity AA-FDEEECPP-FCS_CKM.4-A-ASE-01

The evaluator shall verify the TSS provides a high level description of how keys stored in volatile memory are destroyed. The valuator to verify that TSS outlines:

- if and when the TSF or the Operational Environment is used to destroy keys from volatile memory;

- if and how memory locations for (temporary) keys are tracked;

- details of the interface used for key erasure when relying on the OE for memory clearing.

Summary

Section 7.1.1.4 of the [ST] provides a high-level description of how keys stored in volatile memory are destroyed:

- Keys in volatile memory are destroyed by the Secure Enclave, part of the TOE
- The Secure Enclave tracks the memory location of keys in volatile memory until the relevant cryptographic operation is completed
- The TOE does not rely on the OE for memory clearing, it erases the key material by overwriting it with zeroes

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-A-AGD-01

Assurance Activity AA-FDEEECPP-FCS_CKM.4-A-AGD-01

The evaluator shall check the guidance documentation if the TOE depends on the Operational Environment for memory clearing and how that is achieved.

Summary

The [ST] states that the erase operation is performed by the Secure Enclave (part of the TOE). Therefore, this assurance activity is not applicable.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-A-AKM-01

Assurance Activity AA-FDEEECPP-FCS_CKM.4-A-AKM-01

The evaluator shall check to ensure the KMD lists each type of key, its origin, possible memory

locations in volatile memory.

Summary

Table 1 of the [KMD] lists each type of key, its memory type (volatile or non-volatile) and a description including its origin.

Test Assurance Activities

No assurance activities defined.

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

TSS + Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.4-B-ASE-01

The evaluator examines the TSS to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in non-volatile memory) and how they are overwritten.

The evaluator shall check to ensure the TSS lists each type of key that is stored, and identifies the memory type where key material is stored. When listing the type of memory employed, the TSS will list each type of memory selected in the FCS_CKM.4.1 SFR, as well as any memory types that employ a different memory controller or storage algorithm. For example, if a TOE uses NOR flash and NAND flash, both types are to be listed.

The evaluator shall examine the TSS to ensure it describes the method that is used by the memory controller to write and read memory from each type of memory listed. The purpose here is to provide a description of how the memory controller works so one can determine exactly how keys are written to memory. The description would include how the data is written to and read from memory (e.g., block level, cell level), mechanisms for copies of the key that could potentially exist (e.g., a copy with parity bits, a copy without parity bits, any mechanisms that are used for redundancy).

The evaluator shall examine the TSS to ensure it describes the destruction procedure for each key that has been identified. If different types of memory are used to store the key(s), the evaluator shall check to ensure that the TSS identifies the destruction procedure for each memory type where keys are stored (e.g., key X stored in flash memory is destroyed by overwriting once with zeros, key X' stored in EEPROM is destroyed by a overwrite consisting of a pseudo random pattern – the EEPROM used in the TOE uses a wear-leveling scheme as described).

If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.

Upon completion of the TSS examination, the evaluator understands how all the keys (and potential copies) are destroyed.

Summary

Section 7.1.1.4 of the [ST] describes that keys are introduced in the volatile memory of the Secure Enclave when required for cryptographic operations; they are unwrapped from their wrapped form stored in non-volatile memory. The TSS lists the type of memory used, DRAM for volatile memory, and how data is read from/written to this memory.

The evaluator also verified that the TSS describes the destruction procedures. The Secure Enclave UID cannot be destroyed, as it is fused into the Secure Enclave.

Guidance Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.4-B-AGD-01

There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information. The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, it is assumed the drive supports the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.

Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

It is assumed that if a RAID array is being used, only set-ups that support TRIM are utilized. It is assumed if the drive is connected via PCI-Express, the operating system supports TRIM over that channel. It is assumed the drive is healthy and contains minimal corrupted data and will be end of life before a significant amount of damage to drive health occurs, it is assumed there is a risk small amounts of potentially recoverable data may remain in damaged areas of the drive.

Finally, it is assumed the keys are not stored using a method that would be inaccessible to TRIM, such as being contained in a file less than 982 bytes which would be completely contained in the master file table.

For destruction on wear-leveled memory, if a time period is required before is processed destruction the ST author shall provide an estimated range.

Summary

Section 1.1 of the [CCGUIDE] states that the TOE hardware consists of Apple silicon SoCs which are custom silicon in Mac computers. This ensures that the key destruction requirements are always met and not delayed or wear-leveled.

Test Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM.4-B-ATE-01

For these tests the evaluator shall utilize appropriate development environment (e.g. a Virtual Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

For destruction on wear-leveled memory, if a time period is required before is evaluator shall wait that amount of time after clearing the key in tests 2 and 3.

Test 1: Applied to each key held as plaintext in volatile memory and subject to destruction by overwrite by the TOE (whether or not the plaintext value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator shall:

1. *Record the value of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*

4. *Cause the TOE to stop the execution but not exit.*
5. *Cause the TOE to dump the entire memory of the TOE into a binary file.*
6. *Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.*
7. *Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.*

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a miniscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

Test 2: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

1. *Record the value of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Search the non-volatile memory the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.*
5. *Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for test 1 above), and if a fragment is found in the repeated test then the test fails.*

Test 3: Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

1. *Record the storage location of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Read the storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.*

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

Summary

The evaluator used a tool provided by the vendor to read the wrapped data encryption keys loaded by the Secure Enclave. The evaluator verified that these values changed when the TOE was factory reset, implying that new data encryption keys were generated during factory reset and the old cryptographic keys were destroyed.

This test procedure covered both volatile and non-volatile keys.

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

TSS + Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-D-ASE-01

The evaluator examines the TSS to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in non-volatile memory) and how they are overwritten.

The evaluator shall check to ensure the TSS lists each type of key that is stored in in non-volatile memory, and identifies how the TOE interacts with the underlying platform to manage keys (e.g., store, retrieve, destroy). The description includes details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys (e.g., file system APIs, platform key store APIs).

The evaluator examines the interface description for each different media type to ensure that the interface supports the selection(s) and description in the TSS.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement. If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.

Summary

Section 7.1.1.4 of the [ST] describes that keys are introduced in the volatile memory of the Secure Enclave when required for cryptographic operations; they are unwrapped from their wrapped form stored in non-volatile memory. As the Secure Enclave is part of the TOE hardware, the platform is not used to manage keys.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-D-AGD-01

There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information. The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, it is assumed the drive supports the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.

Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

It is assumed that if a RAID array is being used, only set-ups that support TRIM are utilized. It is assumed if the drive is connected via PCI-Express, the operating system supports TRIM over that channel. It is assumed the drive is healthy and contains minimal corrupted data and will be end of life before a significant amount of damage to drive health occurs, it is assumed there is a risk small amounts of potentially recoverable data may remain in damaged areas of the drive.

Finally, it is assumed the keys are not stored using a method that would be inaccessible to TRIM, such as being contained in a file less than 982 bytes which would be completely contained in the master file table.

Summary

Section 1.1 of the [CCGUIDE] states that the TOE hardware consists of Apple silicon SoCs which are custom silicon in Mac computers. This ensures that the key destruction requirements are always met and not delayed or wear-leveled.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM.4-D-ATE-01

[TD0766]

Test 1: Applied to each key held as plaintext in volatile memory and subject to destruction by overwrite by the TOE (whether or not the plaintext value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator shall:

1. *Record the value of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Cause the TOE to stop the execution but not exit.*
5. *Cause the TOE to dump the entire memory of the TOE into a binary file.*
6. *Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.*
7. *Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.*

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a miniscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

The following tests apply only to for the selection of “logically addresses the storage location...”, since the TOE in this instance has more visibility into what is happening within the underlying platform (e.g., a logical view of the media). For the selection of “instructs the underlying platform...”, the TOE has no visibility into the inner workings and completely relies on the underlying platform, so there is no reason to test the TOE beyond test 1.

For the selection of “logically addresses the storage location...”, the following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern.

Test 2: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media (e.g., MBR file system):

1. *Record the value of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.*
5. *Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for Use Case 1 test 1 above), and if a fragment is found in the repeated test then the test fails.*

Test 3: Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media:

1. *Record the logical storage location of the key in the TOE subject to clearing.*

2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.*

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

Summary

The evaluator used a tool provided by the vendor to read the wrapped data encryption keys loaded by the Secure Enclave. The evaluator verified that these values changed when the TOE was factory reset, implying that new data encryption keys were generated during factory reset and the old cryptographic keys were destroyed.

This test procedure covered both volatile and non-volatile keys.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM_EXT.4-A-ASE-01

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.4-A-ASE-01

The evaluator shall verify the TSS provides a high level description of what it means for keys and key material to be no longer needed and when then should be expected to be destroyed.

Summary

Section 7.1.1.4 of the [ST] provides a high-level description of when key material is expected to be destroyed: keys are destroyed by the Secure Enclave when no longer required to perform a specific cryptographic operation.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM_EXT.4-A-AKM-01

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.4-A-AKM-01

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside and when the keys and key material are no longer needed.

The evaluator shall verify the KMD includes a key lifecycle, that includes a description where key material reside, how the key material is used, how it is determined that keys and key material are no longer needed, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(a) for the destruction.

Summary

Section 3.1 of the [KMD] explains where keys are stored, when keys are loaded in volatile memory, when keys are no longer needed, and how keys are destroyed from volatile memory. The evaluator verified that this documentation follows FCS_CKM.4(a) for the destruction.

Test Assurance Activities

No assurance activities defined.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM_EXT.4-B-ASE-01

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.4-B-ASE-01

The evaluator shall verify the TSS provides a description of what keys and key material are destroyed when entering any Compliant power saving state.

Summary

Section 7.1.1.4 of the [ST] explains that all keys stored in plaintext are destroyed when entering a Compliant power saving state.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM_EXT.4-B-AGD-01

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.4-B-AGD-01

The evaluator shall validate that guidance documentation contains clear warnings and information on conditions in which the TOE may end up in a non-Compliant power saving state indistinguishable from a Compliant power saving state. In that case it must contain mitigation instructions on what to do in such scenarios.

Summary

Section 3.2 of the [CCGUIDE] explains that only the G2(S5) state is a Compliant power saving state. The low power Sleep state supported by macOS is not Compliant and must be disabled using the instructions provided in this section.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_CKM_EXT.4-B-AKM-01

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside.

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(d) for the destruction.

Summary

Section 3.1 of the [KMD] explains where keys are stored, when keys are loaded in volatile memory, when keys are no longer needed, and how keys are destroyed from volatile memory. The evaluator verified that this documentation follows FCS_CKM.4(d) for the destruction.

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.4-B-AKM-01

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside.

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM_EXT.6 for the destruction.

Summary

Section 3.1 of the [KMD] explains where keys are stored, when keys are loaded in volatile memory, when keys are no longer needed, and how keys are destroyed from volatile memory. The evaluator verified that this documentation follows FCS_CKM_EXT.6 for the destruction.

Test Assurance Activities

No assurance activities defined.

2.2.1.10 Cryptographic Key Destruction Types (FCS_CKM_EXT.6)

TSS + Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_CKM_EXT.6-ASE-01

The evaluator shall examine the TOE's keychain in the TSS/KMD and verify all keys subject to destruction are destroyed according to one of the specified methods.

Summary

Section 7.1.1.4 of the [ST] explains that all keys stored in plaintext are destroyed by removing power from the volatile memory.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-A-ASE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-A-ASE-01

The evaluator shall check the TSS to ensure that it describes the overall flow of the signature verification. This should at least include identification of the format and general location (e.g., "firmware on the hard drive device" rather than "memory location 0x00007A4B") of the data to be used in verifying the digital signature; how the data received from the operational environment are brought on to the device; and any processing that is performed that is not part of the digital signature algorithm (for instance, checking of certificate revocation lists).

Summary

Section 7.1.1.5 of the [ST] describes the overall flow of the signature verification, including the format and general location of the data (firmware/software updates and secure boot), and how the data is received (during installation or the boot procedure). There is no processing performed that is not part of the signature algorithm.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-A-ATE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-A-ATE-01

[TD1042] Each section below contains the tests the evaluators must perform for each type of digital signature scheme. Based on the assignments and selections in the requirement, the evaluators choose the specific activities that correspond to those selections.

It should be noted that for the schemes given below, there are no key generation/domain parameter generation testing requirements. This is because it is not anticipated that this functionality would be needed in the end device, since the functionality is limited to checking digital signatures in delivered updates. This means that the domain parameters should have already been generated and encapsulated in the hard drive firmware or on-board non-volatile storage. If key generation/domain parameter generation is required, the evaluation and validation scheme must be consulted to ensure the correct specification of the required evaluation activities and any additional

components.

The following tests are conditional based upon the selections made within the SFR.

The following tests may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

ECDSA Algorithm Tests

ECDSA FIPS 186-5 Signature Verification Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

RSA Signature Algorithm Tests

Signature Verification Test

The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's authentic and unauthentic signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.

The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.

Summary

The evaluator verified that Automated Cryptographic Validation Testing System (ACVTS) is used to test all cryptographic algorithms in accordance with the CAVP test procedures. The results have been validated by the CAVP for the signature generation and verification algorithms, and the certificate information is provided in Section 2.1 of this document.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-B-ASE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-B-ASE-01

The evaluator shall check that the association of the hash function with other TSF cryptographic

functions (for example, the digital signature verification function) is documented in the TSS.

Summary

Section 7.1.1.6 of the [ST] indicates that the hash functions are used for PBKDF2 and digital signature verification.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-B-AGD-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-B-AGD-01

The evaluator checks the operational guidance documents to determine that any system configuration necessary to enable required hash size functionality is provided.

Summary

Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-B-ATE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-B-ATE-01

The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented test mode.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this cPP.

Short Messages Test Bit-oriented Mode

The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test Byte-oriented Mode

The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test Bit-oriented Mode

*The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 99*i, where 1 ≤ i ≤ m. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

Selected Long Messages Test Byte-oriented Mode

*The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

Pseudorandomly Generated Messages Test

This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of the NIST Secure Hash Algorithm Validation System (SHAVS) (https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Algorithm-Validation-Program/documents/shs/SHAVS.pdf). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Summary

The evaluator verified that Automated Cryptographic Validation Testing System (ACVTS) is used to test all cryptographic algorithms in accordance with the CAVP test procedures. The results have been validated by the CAVP for the hash algorithms, and the certificate information is provided in Section 2.1 of this document.

2.2.1.13 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(c))

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-C-ASE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-C-ASE-01

If HMAC was selected:

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

If CMAC was selected:

The evaluator shall examine the TSS to ensure that it specifies the following values used by the CMAC function: key length, block cipher used, block size (of the cipher), and output MAC length used.

Summary

Section 7.1.1.7 of the [ST] contains the key length, hash function used, block size, and output MAC length used by the HMAC function.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-C-ATE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-C-ATE-01

If HMAC was selected:

For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key using a known good implementation.

If CMAC was selected:

For each of the supported parameter sets, the evaluator shall compose at least 15 sets of test data. Each set shall consist of a key and message data. The test data shall include messages of different lengths, some with partial blocks as the last block and some with full blocks as the last block. The test data keys shall include cases for which subkey K1 is generated both with and without using the irreducible polynomial R_b, as well as cases for which subkey K2 is generated from K1 both with and without using the irreducible polynomial R_b. (The subkey generation and polynomial R_b are as defined in SP800-38E.) The evaluator shall have the TSF generate CMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating CMAC tags with the same key using a known good implementation.

Summary

The evaluator verified that Automated Cryptographic Validation Testing System (ACVTS) is used to test all cryptographic algorithms in accordance with the CAVP test procedures. The results have been validated by the CAVP for the keyed-hash message authentication algorithms, and the certificate information is provided in Section 2.1 of this document.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-D-ASE-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-D-ASE-01

The evaluator shall verify the TSS includes a description of the key wrap function(s) and shall verify the key wrap uses an approved key wrap algorithm according to the appropriate specification.

Summary

Section 7.1.1.8 of the [ST] contains a description of the key wrap function (AES in KW mode), according to SP 800-38F.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-D-AKM-01

Assurance Activity AA-FDEEECPP-FCS_COP.1-D-AKM-01

The evaluator shall review the KMD to ensure that all keys are wrapped using the approved method and a description of when the key wrapping occurs.

Summary

Section 2.2 of the [KMD] explains that the DEKs and KEKs are wrapped using AES-KW. This is also illustrated in Figure 2 of the [KMD].

Test Assurance Activities

No assurance activities defined.

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

TSS Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_COP.1-F-ASE-01

The evaluator shall verify the TSS includes a description of the key size used for encryption and the mode used for encryption.

Summary

Section 7.1.1.9 of the [ST] contains a description of the key size and encryption mode (AES in XTS mode with a 256-bit key).

Guidance Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_COP.1-F-AGD-01

If multiple encryption modes are supported, the evaluator examines the guidance documentation to determine that the method of choosing a specific mode/key size by the end user is described

Summary

Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_COP.1-F-ATE-01

The following tests are conditional based upon the selections made in the SFR.

AES-CBC Tests

For the AES-CBC tests described below, the plaintext, ciphertext, and IV values shall consist of 128-bit blocks. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known-good implementation.

These tests are intended to be equivalent to those described in NIST's AES Algorithm Validation Suite (AESAVS) (http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf). Known answer values tailored to exercise the AES-CBC implementation can be obtained using NIST's CAVS Algorithm Validation Tool or from NIST's ACPV service for automated algorithm tests (acvp.nist.gov), when available. It is not recommended that evaluators use values obtained from static sources such as the example NIST's AES Known Answer Test Values from the AESAVS document, or use values not generated expressly to exercise the AES-CBC implementation.

AES-CBC Known Answer Tests

KAT-1 (GFSBox):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different plaintext values for each selected key size and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros.

To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different ciphertext values for each selected key size and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using a key value of all zeros and an IV of all zeros.

KAT-2 (KeySBox):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros.

To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size and obtain the plaintext that results from AES-CBC decryption of an all-zeros ciphertext using the given key and an IV of all zeros.

KAT-3 (Variable Key):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of keys for each selected key size (as described below) and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using each key and an IV of all zeros.

Key i in each set shall have the leftmost i bits set to ones and the remaining bits to zeros, for values of i from 1 to the key size. The keys and corresponding ciphertext are listed in AESAVS, Appendix E.

To test the decrypt functionality of AES-CBC, the evaluator shall use the same keys as above to decrypt the ciphertext results from above. Each decryption should result in an all-zeros plaintext.

KAT-4 (Variable Text):

To test the encrypt functionality of AES-CBC, for each selected key size, the evaluator shall supply a set of 128-bit plaintext values (as described below) and obtain the ciphertext values that result from AES-CBC encryption of each plaintext value using a key of each size and IV consisting of all zeros.

Plaintext value i shall have the leftmost i bits set to ones and the remaining bits set to zeros, for values of i from 1 to 128. The plaintext values are listed in AESAVS, Appendix D.

To test the decrypt functionality of AES-CBC, for each selected key size, use the plaintext values from above as ciphertext input, and AES-CBC decrypt each ciphertext value using key of each size consisting of all zeros and an IV of all zeros.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a plaintext message of length i blocks, and encrypt the message using AES-CBC. The resulting ciphertext values shall be compared to the results of encrypting the plaintext messages using a known good implementation.

The evaluator shall test the decrypt functionality by decrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a ciphertext message of length i blocks, and decrypt the message using AES-CBC. The resulting plaintext values shall be compared to the results of decrypting the ciphertext messages using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality for each selected key size using 100 3-tuples of pseudo-random values for plaintext, IVs, and keys.

The evaluator shall supply a single 3-tuple of pseudo-random values for each selected key size. This 3-tuple of plaintext, IV, and key is provided as input to the below algorithm to generate the remaining 99 3-tuples, and to run each 3-tuple through 1000 iterations of AES-CBC encryption.

```
# Input: PT, IV, Key
Key[0] = Key
IV[0] = IV
PT[0] = PT
for i = 1 to 100 {
      Output Key[i], IV[i], PT[0]
```

```
            for j = 1 to 1000 {
            if j == 1 {
                  CT[1] = AES-CBC-Encrypt(Key[i], IV[i], PT[1])
                  PT[2] = IV[i]
            } else {
                  CT[j] = AES-CBC-Encrypt(Key[i], PT[j])
                  PT[j+1] = CT[j-1]
            }
      }
      Output CT[1000]
      If KeySize == 128 { Key[i+1] = Key[i] xor CT[1000] }
      If KeySize == 256 { Key[i+1] = Key[i] xor ((CT[999] << 128) |
      CT[1000]) }
      IV[i+1] = CT[1000]
      PT[0] = CT[999]
}
```

The ciphertext computed in the 1000th iteration (CT[1000]) is the result for each of the 100 3-tuples for each selected key size. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as above, exchanging CT and PT, and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Test

The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

128 bit and 256 bit keys

Two plaintext lengths. *One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.*

Three AAD lengths. *One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.*

Two IV lengths. *If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.*

The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

XTS-AES Test

The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

256 bit (for AES-128) and 512 bit (for AES-256) keys

Three data unit (i.e., plaintext) lengths*. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.*

using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

Summary

An ACVP test harness extension was developed by the lab to perform ACVP-like tests for AES-XTS with internally generated tweak values. This tool compares the output generated by the TOE with the known good output of OpenSSL 3.6.0. The evaluator used this tool to verify the output generated by the TOE is identical to the output of OpenSSL 3.6.0, which confirms the correct implementation of AES-XTS in the TOE.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-G-ASE-01

[TD1046] The evaluator shall verify the TSS includes a description of the key size used for encryption and the mode used for the key encryption.

If AES-XTS is claimed then the evaluator shall examine the TSS to verify that the TOE creates full-length keys by methods that ensure that the two key halves are distinct. The evaluator shall confirm the TSS describes the block of data containing the key and that is full block of data in alignment with the selected AES standard.

Summary

Section 7.1.1.10 of the [ST] contains a description of the key size and encryption mode (AES in CBC mode with a 256-bit key). AES-XTS is not claimed.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-G-AGD-01

If multiple key encryption modes are supported, the evaluator examines the guidance documentation to determine that the method of choosing a specific mode/key size by the end user is described

Summary

Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-G-AKM-01

The evaluator shall examine the vendor's KMD to verify that it includes a description of how key encryption will be used as part of the key chain.

Summary

Section 2.1 of the [KMD] explains that AES-CBC is used to derive the Password-Derived Key (PDK) from the password authentication factor.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_COP.1-G-ATE-01

The AES test should be followed in FCS_COP.1(f) Cryptographic Operation (AES Data Encryption/Decryption.

Summary

The evaluator verified that Automated Cryptographic Validation Testing System (ACVTS) is used to test all cryptographic algorithms in accordance with the CAVP test procedures. The results have been validated by the CAVP for the key encryption algorithms, and the certificate information is provided in Section 2.1 of this document.

2.2.1.17 Key Chaining (Initiator) (FCS_KYC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_KYC_EXT.1-ASE-01

The evaluator shall verify the TSS contains a high-level description of the BEV sizes – that it supports BEV outputs of no fewer 128 bits for products that support only AES-128, and no fewer than 256 bits for products that support AES-256.

Summary

Section 7.1.1.11 of the [ST] states the TOE only supports 256 bit BEVs.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_KYC_EXT.1-AKM-01

The evaluator shall examine the KMD describes a high level description of the key hierarchy for all authorizations methods selected in FCS_AFA_EXT.1 that are used to protect the BEV. The evaluator shall examine the KMD to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap or key derivation methods that meet FCS_COP.1(d) and FCS_KDF_EXT.1.

The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the effective strength of the BEV is maintained throughout the key chain.

The evaluator shall verify the KMD includes a description of the strength of keys throughout the key chain.

Summary

Section 2.1 of the [KMD] provides a detailed description of the Password-Derived Key (BEV) hierarchy and the key chain. This description also includes a diagram illustrating the key chain (Figure 1 of the [KMD]) and states that the strength of all keys (UID, System Measurement Device Key, Password-Derived Key) involved in the chain is 256 bits. The evaluator verified that this strength is maintained throughout the key chain.

Test Assurance Activities

No assurance activities defined.

2.2.1.18 Key Chaining (Recipient) (FCS_KYC_EXT.2)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FCS_KYC_EXT.2-AKM-01

The evaluator shall examine the KMD to ensure it describes a high level key hierarchy and details of the key chain. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap or key derivation methods that meet FCS_KDF_EXT.1, FCS_COP.1(d), FCS_COP.1(e), and/or FCS_COP.1(g).

The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or knowledge of the BEV and the effective strength of the DEK is maintained throughout the Key Chain.

The evaluator shall verify the KMD includes a description of the strength of keys throughout the key chain.

Summary

Section 2.2 of the [KMD] provides a detailed description of the key chain from the Password-Derived Key (BEV) to the Volume Encryption Key (DEK), including the Media Key, Key Encryption Key, and Root Encryption Key. This description also includes a diagram illustrating the key chain (Figure 2 of the [KMD]) and states that the strength of all keys involved in the chain is 256 bits.

The evaluator verified that this strength is maintained throughout the key chain, and in particular the effective strength of the Volume Encryption Key is maintained.

Test Assurance Activities

No assurance activities defined.

2.2.1.19 Cryptographic Password Construct and Conditioning (FCS_PCC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_PCC_EXT.1-ASE-01

The evaluator shall ensure the TSS describes the manner in which the TOE enforces the construction of passwords, including the length, and requirements on characters (number and type). The evaluator also verifies that the TSS provides a description of how the password is conditioned and the evaluator ensures it satisfies the requirement.

Summary

Section 7.1.1.1 of the [ST] describes that the TOE allows passwords of up to and including 255 characters consisting of any combination of uppercase characters, lowercase characters, numbers, and any other 8-bit special character. Furthermore, this section explains that the password is conditioned using PBKDF2 with HMAC-SHA-256, then repeatedly encrypted using AES-CBC.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_PCC_EXT.1-AKM-01

The evaluator shall examine the KMD to ensure that the formation of the BEV and intermediary keys is described and that the key sizes match that selected by the ST author.

The evaluator shall check that the KMD describes the method by which the password/passphrase is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself. The evaluator shall verify that the KMD contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length as the BEV as specified above.

Summary

Section 2.1 of the [KMD] describes the inputs to PBKDF2: the password is provided without any pre-processing, a 128-bit salt is randomly generated, and the iteration count is set to 1. This section also states that the resulting value is repeatedly encrypted using AES-CBC-256 with the 256-bit Secure Enclave Unique ID. The evaluator confirmed all key sizes matched the [ST] selections.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_PCC_EXT.1-ATE-01

The evaluator shall also perform the following tests:

- *Test 1: Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.*
- *Test 2: If the TOE supports a password/passphrase length up to a maximum number of characters, n (which would be greater than 64), then ensure that the TOE will not accept more than n characters.*
- *Test 3: Ensure that the TOE supports passwords consisting of all characters assigned and supported by the ST author.*

Summary

Test 1: The evaluator changed the password to a password of length 64 characters. The evaluator verified that authentication with the 64-character password was successful.

Test 2: The TOE supports passwords up to 255 characters. The evaluator attempted to change the password to a password of length 256 characters, and observed the password change failed.

Test 3: The TOE supports passwords consisting of all 8-bit values. The evaluator changed the password to a password consisting of uppercase and lowercase alphabetic, numeric, and various special characters. The evaluator verified that authentication with new password was successful.

2.2.1.20 Random Bit Generation (FCS_RBG_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_RBG_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FCS_RBG_EXT.1-ASE-01

For any RBG services provided by a third party, the evaluator shall ensure the TSS includes a statement about the expected amount of entropy received from such a source, and a full description of the processing of the output of the third-party source. The evaluator shall verify that this statement is consistent with the selection made in FCS_RBG_EXT.1.2 for the seeding of the DRBG. If the ST specifies more than one DRBG, the evaluator shall examine the TSS to verify that it identifies the usage of each DRBG mechanism.

Summary

Section 7.1.1.12 of the [ST] states that the TOE performs random bit generation (as part of the Secure Enclave TRNG). The [ST] specifies only one DRBG.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_RBG_EXT.1-AGD-01

Assurance Activity AA-FDEEECPP-FCS_RBG_EXT.1-AGD-01

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected DRBG mechanism(s), if necessary, and provides information regarding how to instantiate/call the DRBG for RBG services needed in this cPP.

Summary

Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_RBG_EXT.1-ATE-01

Assurance Activity AA-FDEEECPP-FCS_RBG_EXT.1-ATE-01

The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable by the TOE, the evaluator shall perform 15 trials for each configuration. The evaluator shall verify that the instructions in the operational guidance for configuration of the RNG are valid.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate

the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

Summary

The evaluator verified that Automated Cryptographic Validation Testing System (ACVTS) is used to test all cryptographic algorithms in accordance with the CAVP test procedures. The results have been validated by the CAVP for the DRBG algorithms, and the certificate information is provided in Section 2.1 of this document.

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

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_SNI_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FCS_SNI_EXT.1-ASE-01

[TD0760]

If salts are used, the evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generating using an RBG described in FCS_RBG_EXT.1 or by the Operational Environment. If external function is used for this purpose, the TSS should include the specific API that is called with inputs.

If IVs or nonces are used, the evaluator shall ensure the TSS describes how nonces are created uniquely and how IVs and tweaks are handled (based on the AES mode). The evaluator shall confirm that the nonces are unique and the IVs and tweaks meet the stated requirements.

Summary

Section 7.1.1.13 of the [ST] describes that the TOE generates salts and initialization vectors (IVs) using the TRNG. For AES-XTS the tweak value is the physical block number of the media on which the file is being written.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.2.1.22 Validation (FCS_VAL_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_VAL_EXT.1-ASE-01

The evaluator shall examine the TSS to determine which authorization factors support validation.

The evaluator shall examine the TSS to review a high-level description if multiple submasks are used within the TOE, how the submasks are validated (e.g., each submask validated before combining, once combined validation takes place).

Summary

Section 7.1.1.14 of the [ST] states that a password is the only supported authorization factor and the password is used to derive the PDK, the only submask used within the TOE. The PDK is validated using a hardware-enforced “counter lockbox”: a password verifier value is derived, which must match the stored password verifier value.

Assurance Activity AA-FDEEECPP-FCS_VAL_EXT.1-ASE-01

The evaluator shall examine the TSS to determine which authorization factors support validation.

The evaluator shall examine the TSS to review a high-level description if multiple submasks are used within the TOE, how the submasks are validated (e.g., each submask validated before combining, once combined validation takes place).

The evaluator shall also examine the TSS to determine that a subset or all of the authorization factors identified in the SFR can be used to exit from a Compliant power saving state.

Summary

Section 7.1.1.14 of the [ST] states that a password is the only supported authorization factor and the password is used to derive the PDK, the only submask used within the TOE. The PDK is validated using a hardware-enforced “counter lockbox”: a password verifier value is derived, which must match the stored password verifier value.

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

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_VAL_EXT.1-AGD-01

(conditional) If the validation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established.

(conditional) If the validation functionality is specified by the ST author, the evaluator shall examine

the operational guidance to ensure that it states the values that the TOE uses for limits regarding validation attempts.

Summary

The validation functionality is not configurable. Section 4 of the [CCGUIDE] describes the limits on the password validation attempts.

Assurance Activity AA-FDEEECPP-FCS_VAL_EXT.1-AGD-01

(conditional) If the validation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established.

(conditional) If the validation functionality is specified by the ST author, the evaluator shall examine the operational guidance to ensure that it states the values that the TOE uses for limits regarding validation attempts.

The evaluator shall verify that the guidance documentation states which authorization factors are allowed to exit a compliant power saving state.

Summary

The validation functionality is not configurable. Section 4 of the [CCGUIDE] describes the limits on the password validation attempts.

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

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_VAL_EXT.1-AKM-01

The evaluator shall examine the KMD to verify that it described the method the TOE employs to limit the number of consecutively failed authorization attempts.

The evaluator shall examine the vendor's KMD to ensure it describes how validation is performed. The description of the validation process in the KMD provides detailed information how the TOE validates the submasks. The KMD describes how the process works, such that it does not expose any material that might compromise the submask(s).

Summary

Section 4.3 of the [KMD] describes how the PDK, the only submask used within the TOE, is validated using a hardware-enforced "counter lockbox": a password verifier value is derived, which must match the stored password verifier value. Additionally, this section explains that the counter from the counter lockbox is used to enforce delays between authentication attempts.

Assurance Activity AA-FDEEECPP-FCS_VAL_EXT.1-AKM-01

The evaluator shall examine the KMD to verify that it described the method the TOE employs to limit the number of consecutively failed authorization attempts.

The evaluator shall examine the vendor's KMD to ensure it describes how validation is performed. The description of the validation process in the KMD provides detailed information how the TOE validates the BEV.

The KMD describes how the process works, such that it does not expose any material that might compromise the submask(s).

Summary

Section 4.3 of the [KMD] describes how the PDK, the only submask used within the TOE, is validated using a hardware-enforced “counter lockbox”: a password verifier value is derived, which must match the stored password verifier value. This derivation happens in the Secure Enclave, part of the TOE hardware, preventing exposure of the password. Additionally, this section explains that the counter from the counter lockbox is used to enforce delays between authentication attempts.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FCS_VAL_EXT.1-ATE-01

The evaluator shall perform the following tests:

Test 1: The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any “lockout” period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.

Test 2: For each validated authorization factor, ensure that when the user provides an incorrect authorization factor, the TOE prevents the BEV from being forwarded outside the TOE (e.g., to the EE).

Summary

Test 1: The evaluator attempted to log in to the TOE using a distinct, incorrect password. After 4 consecutive attempts, a time delay of 1 minute was enforced. The evaluator verified the TOE behaved as described in the TSS.

Test 2: The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator rebooted the TOE and verified that the keys were not loaded if the correct password is not entered.

Assurance Activity AA-FDEEECPP-FCS_VAL_EXT.1-ATE-01

The evaluator shall perform the following tests:

Test 1: The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any “lockout” period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.

Test 2: The evaluator shall force the TOE to enter a Compliant power saving state, attempt to resume it from this state, and verify that only a valid authorization factor as defined by the guidance documentation is sufficient to allow the TOE to exit the Compliant power saving state.

Summary

Test 1: The evaluator attempted to log in to the TOE using a distinct, incorrect password. After 4 consecutive attempts, a time delay of 1 minute was enforced. The evaluator verified the TOE behaved as described in the TSS.

Test 2: The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE and verified that the keys were not loaded if the correct password is not entered. Then, the evaluator logged in using the password and verified the keys were available. This proves that the BEV is forwarded if and only if a correct authorization factor is provided.

2.2.2 User data protection (FDP)

2.2.2.1 Protection of Data on Disk (FDP_DSK_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEEECPP-FDP_DSK_EXT.1-ASE-01

The evaluator shall examine the TSS to ensure that the description is comprehensive in how the data is written to the disk and the point at which the encryption function is applied. The TSS must make the case that standard methods of accessing the disk drive via the host platforms operating system will pass through these functions.

For the cryptographic functions that are provided by the Operational Environment, the evaluator shall check the TSS to ensure it describes, for each platform identified in the ST, the interface(s) used by the TOE to invoke this functionality.

The evaluator shall verify the TSS in performing the evaluation activities for this requirement. The evaluator shall ensure the comprehensiveness of the description, confirms how the TOE writes the data to the disk drive, and the point at which it applies the encryption function.

The evaluator shall verify that the TSS describes the initialization of the TOE and the activities the TOE performs to ensure that it encrypts all the storage devices entirely when a user or administrator first provisions the TOE. The evaluator shall verify the TSS describes areas of the disk that it does not encrypt (e.g., portions associated with the Master Boot Records (MBRs), boot loaders, partition tables, etc.). If the TOE supports multiple disk encryptions, the evaluator shall examine the administration guidance to ensure the initialization procedure encrypts all storage devices on the platform.

Summary

Section 7.1.2.1 of the [ST] contains a detailed description of how data is written to the disk and when encryption is applied. The data is written by the DMA Storage Controller, which performs the encryption/decryption prior to reaching the actual storage (or application processor). There is no way for the host platform to write/read data without passing through the DMA Storage Controller. All cryptography is performed internal to the TOE, as the DMA Storage Controller is part of the TOE hardware.

The evaluator also verified that the TSS describes when encryption is enabled and the areas of the disk that are not encrypted.

Guidance Assurance Activities

Assurance Activity AA-FDEEECPP-FDP_DSK_EXT.1-AGD-01

The evaluator shall review the AGD guidance to determine that it describes the initial steps needed to enable the FDE function, including any necessary preparatory steps. The guidance shall provide instructions that are sufficient, on all platforms, to ensure that all hard drive devices will be encrypted when encryption is enabled.

Summary

Section 3.1 of the [CCGUIDE] states that FileVault is installed with macOS, but it must be enabled by the administrator. The evaluator verified that the instructions provided in this section are sufficient to enable the FDE function.

Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FDP_DSK_EXT.1-AKM-01

The evaluator shall verify the KMD includes a description of the data encryption engine, its components, and details about its implementation (e.g. for hardware: integrated within the device's

main SOC or separate co-processor, for software: initialization of the product, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and areas which are not encrypted (e.g. boot loaders, portions associated with the Master Boot Record (MBRs), partition tables, etc.)). The evaluator shall verify the KMD provides a functional (block) diagram showing the main components (such as memories and processors) and the data path between, for hardware, the device's host interface and the device's persistent media storing the data, or for software, the initial steps needed to the activities the TOE performs to ensure it encrypts the storage device entirely when a user or administrator first provisions the product. The hardware encryption diagram shall show the location of the data encryption engine within the data path. The evaluator shall validate that the hardware encryption diagram contains enough detail showing the main components within the data path and that it clearly identifies the data encryption engine.

The evaluator shall verify the KMD provides sufficient instructions for all platforms to ensure that when the user enables encryption, the product encrypts all hard storage devices. The evaluator shall verify that the KMD describes the data flow from the device's host interface to the device's persistent media storing the data. The evaluator shall verify that the KMD provides information on those conditions in which the data bypasses the data encryption engine (e.g. read-write operations to an unencrypted Master Boot Record area).

The evaluator shall verify that the KMD provides a description of the platform's boot initialization, the encryption initialization process, and at what moment the product enables the encryption. The evaluator shall validate that the product does not allow for the transfer of user data before it fully initializes the encryption. The evaluator shall ensure the software developer provides special tools which allow inspection of the encrypted drive either in-band or out-of-band, and may allow provisioning with a known key.

Summary

Section 4.2 of the [KMD] describes the data encryption engine used by the TOE. The evaluator verified that this description includes the components (DMA Storage Controller), implementation (integrated in the Apple silicon SoC), logical interfaces (read / write data path), and areas that are not encrypted. Additionally, this description includes a diagram (Figure 3) showing the main components and the data path. The evaluator verified this diagram is sufficiently detailed and clearly identifies the data encryption engine.

The evaluator verified that the aforementioned section includes details regarding the encryption of hard storage devices, data flow through the DMA Storage Controller, data bypassing the data encryption engine, and platform initialization (immediately after start-up).

There are currently no tools available to check the encrypted data in the storage drive, nor can the encryption/decryption functionality for data be bypassed.

Test Assurance Activities

Assurance Activity AA-FDEEECPP-FDP_DSK_EXT.1-ATE-01

The evaluator shall perform the following tests:

Test 1: Write data to random locations, perform required actions and compare:

- *Ensure TOE is initialized and, if hardware, encryption engine is ready;*
- *Provision TOE to encrypt the storage device. For SW Encryption products, or hybrid products use a known key and the developer tools.*
- *Determine a random character pattern of at least 64 KB;*
- *Retrieve information on what the device TOE's lowest and highest logical address is for which encryption is enabled.*

Test 2: Write pattern to storage device in multiple locations:

- *For HW Encryption, randomly select several logical address locations within the device's*

lowest to highest address range and write pattern to those addresses;

- *For SW Encryption, write the pattern using multiple files in multiple logical locations.*

Test 3: Verify data is encrypted:

- *For HW Encryption:*
  - *engage device's functionality for generating a new encryption key, thus performing an erase of the key per FCS_CKM.4(a);*
  - *Read from the same locations at which the data was written;*
  - *Compare the retrieved data to the written data and ensure they do not match*
- *For SW Encryption, using developer tools;*
  - *Review the encrypted storage device for the plaintext pattern at each location where the file was written and confirm plaintext pattern cannot be found.*
  - *Using the known key, verify that each location where the file was written, the plaintext pattern can be correctly decrypted using the key.*
  - *If available in the developer tools, verify there are no plaintext files present in the encrypted range.*

Summary

Test 1: The evaluator successfully created an encrypted partition. In addition, the evaluator created a test file containing 64 KB of random hexadecimal characters, to be used for Test 2 and Test 3.

Test 2: The evaluator copied the test file to the encrypted partition. Then, the evaluator unmounted the partition and exported it as a raw disk dump. The evaluator verified that the random character pattern was not present in the dump of the partition.

Test 3: The evaluator successfully created a second encrypted partition and copied the test file to this partition. Then, the evaluator unmounted the second partition and exported it as a raw disk dump. The evaluator verified that the random character pattern was not present in the second dump. Moreover, the disk dumps of the two test partitions are different, indicating the data is encrypted.

2.2.3 Security management (FMT)

2.2.3.1 Management of Functions Behavior (FMT_MOF.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_MOF.1-ASE-01

If support for Compliant power saving state(s) are claimed in the ST, the evaluator shall ensure the TSS describes how these are managed and shall ensure that TSS describes how only privileged users (administrators) are allowed to manage the states.

Summary

Section 7.1.3 of the [ST] states that users cannot modify the behavior of the Compliant power saving state.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_MOF.1-AGD-01

The evaluator to check if guidance documentation describes which authorization factors are required

to change Compliant power saving state behavior and properties.

Summary

Section 3.2 of the [CCGUIDE] states that the Compliant power saving state behavior is not configurable.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_MOF.1-ATE-01

[TD0765]

The evaluator shall perform the following tests:

Test 1 (conditional): If the product supports changes to compliant power saving states, the evaluator presents a privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior and properties are allowed.

Test 2 (conditional): If the product supports changes to compliant power saving states, the evaluator presents a non-privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior are not allowed.

Summary

Test 1: The TOE does not support changes to the Compliant power saving state. Therefore, this test is not applicable.

Test 2: The TOE does not support changes to the Compliant power saving state. Therefore, this test is not applicable.

2.2.3.2 Specification of Management Functions (FMT_SMF.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMF.1-ASE-01

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to change the DEK.

If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to cryptographically erase the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the methods by which users may change the set of all authorization factor values supported.

If item d) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: If power saving states can be managed, the evaluator shall ensure that the TSS describes how this is performed, including how the TOE supports disabling certain power saving states if more than one are supported. If additional management functions are claimed in the ST, the evaluator shall ensure the TSS describes the additional functions.

Summary

Section 7.1.3 of the [ST] describes the management functions:

a) Requests to change the DEK are processed directly by the TOE.

b) Requests to erase the DEK are processed directly by the TOE.

c) Users can change their password using System Settings.

d) Users can initiate the TOE update process using System Settings.

e) Not applicable as the [ST] does not claim additional management functions.

Assurance Activity AA-FDEEECPP-FMT_SMF.1-ASE-01

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE changes the DEK.

If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE cryptographically erases the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: If additional management functions are claimed in the ST, the evaluator shall verify that the TSS describes those functions.

Summary

Section 7.1.3 of the [ST] describes the management functions:

a) The DEK is changed by erasing the appropriate volume using Disk Utility.

b) The DEK is erased by erasing the appropriate volume using Disk Utility.

c) Users can initiate the TOE update process using System Settings.

d) Not applicable as the [ST] does not claim additional management functions.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMF.1-AGD-01

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how the functions for A and B can be initiated by the user.

If item c) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how selected authorization factor values are changed.

If item d) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: Default Authorization Factors: It may be the case that the TOE arrives with default authorization factors in place. If it does, then the selection in section E must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist.

Disable Key Recovery: The guidance for disabling this capability shall be described in the AGD documentation.

Power Saving: The guidance shall describe the power saving states that are supported by the TSF, how these states are applied, how to configure when these states are applied (if applicable), and how to enable/disable the use of specific power saving states (if applicable).

Summary

The evaluator verified that the [CCGUIDE] contains the appropriate descriptions, functions, and instructions:

a) Section 3.4, *Changing/Erasing the Data Encryption Key*

b) Section 3.4, *Changing/Erasing the Data Encryption Key*

c) Section 3.3.3, *Change the Password*

d) Section 2.2, *Update*

e) Not applicable as the [ST] does not claim additional management functions.

Key recovery is not a function provided by the TOE.

Section 3.2 of the [CCGUIDE] describes the supported power saving states and how they are applied. The Compliant power saving state cannot be configured or disabled.

Assurance Activity AA-FDEEECPP-FMT_SMF.1-AGD-01

If item a) is selected in FMT_SMF.1.1: The evaluator shall review the AGD guidance and shall determine that the instructions for changing a DEK exist. The instructions must cover all environments on which the TOE is claiming conformance, and include any preconditions that must exist in order to successfully generate or re-generate the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: Default Authorization Factors: It may be the case that the TOE arrives with default authorization factors in place. If it does, then the selection in item D must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist.

Disable Key Recovery: The guidance for disabling this capability shall be described in the AGD documentation.

Summary

The evaluator verified that the [CCGUIDE] contains the appropriate descriptions, functions, and instructions:

a) Section 3.4, *Changing/Erasing the Data Encryption Key*

b) Section 3.4, *Changing/Erasing the Data Encryption Key*

c) Section 2.2, *Update*

d) Not applicable as the [ST] does not claim additional management functions.

Key recovery is not a function provided by the TOE.

Key Management Assurance Activities

Assurance Activity AA-FDEEECPP-FMT_SMF.1-AKM-01

If item d) is selected in FMT_SMF.1.1: If the TOE offers the functionality to import an encrypted DEK, the evaluator shall ensure the KMD describes how the TOE imports a wrapped DEK and performs the decryption of the wrapped DEK.

Summary

The [ST] does not claim the functionality to import a wrapped DEK. Therefore, this assurance activity is not applicable.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMF.1-ATE-01

[TD0767]

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to forward a command to the EE to change and cryptographically erase the DEK. The actual testing of the cryptographic erase will take place in the EE.

If item c) is selected in FMT_SMF.1.1: The evaluator shall initialize the TOE such that it requires the user to input an authorization factor in order to access encrypted data.

Test 1: The evaluator shall first provision user authorization factors, and then verify all authorization values supported allow the user access to the encrypted data. Then the evaluator shall exercise the management functions to change a user's authorization factor values to a new one. Then he or she will verify that the TOE denies access to the user's encrypted data when he or she uses the old or original authorization factor values to gain access.

If item d) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.

Test 2 (conditional): If the TOE provides default authorization values, the evaluator shall change these values in the course of taking ownership of the device as described in the operational guidance. The evaluator shall then confirm that the (old) authorization values are no longer valid for data access.

Test 3 (conditional): If the TOE provides key recovery capability whose effects are visible at the TOE interface, then the evaluator shall devise a test that ensures that the key recovery capability has been or can be disabled following the guidance provided by the vendor.

Test 4 (conditional): If the TOE provides the ability to configure the power saving states that are entered by certain events, the evaluator shall devise a test that causes the TOE to enter a specific power saving state, configure the TSF so that this activity causes a different state to be entered, repeat the activity, and observe the new state is entered as configured.

Test 5 (conditional): If the TOE provides the ability to disable the use of one or more power saving states, the evaluator shall devise a test that enables all supported power saving states and demonstrates that the TOE can enter into each of these states. The evaluator shall then disable the supported power saving states one by one, repeating the same set of actions that were performed at the start of the test, and observe each time that when a power saving state is configured to no longer be used, none of the behavior causes the disabled state to be entered.

Summary

a) The evaluator verified that the DEK can be changed by erasing the volume using Disk Utility or by factory reset of the TOE.

b) The evaluator verified that the DEK can be erased by erasing the volume using Disk Utility or by factory reset of the TOE.

c) The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first changed the password on the TOE, then rebooted the TOE and attempted to log in using the old password. The evaluator verified the keys were not available. Finally, the evaluator logged in using the new password and verified the keys were available.

   As these keys are used to encrypt the plaintext data, the decrypted plaintext data is only accessible if the keys are available. In other words, the correct (new) password must be supplied to access the decrypted plaintext data.

d) The evaluator used System Settings to verify that the TOE update process can be initiated by a user.

e) Not applicable as the [ST] does not claim additional management functions.

Assurance Activity AA-FDEEECPP-FMT_SMF.1-ATE-01

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to change and cryptographically erase the DEK (effectively removing the ability to retrieve previous user data).

If item c) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.

Summary

a) The evaluator verified that the DEK can be changed by erasing the volume using Disk Utility or by factory reset of the TOE.

b) The evaluator verified that the DEK can be erased by erasing the volume using Disk Utility or by factory reset of the TOE.

c) The evaluator used System Settings to verify that the TOE update process can be initiated by a user.

d) Not applicable as the [ST] does not claim additional management functions.

2.2.3.3 Security Roles (FMT_SMR.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMR.1-ASE-01

There are no TSS evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

Summary

Please refer to the TSS assurance activities for FMT_MOF.1 and FMT_SMF.1.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMR.1-AGD-01

There are no guidance evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

Summary

Please refer to the guidance assurance activities for FMT_MOF.1 and FMT_SMF.1.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FMT_SMR.1-ATE-01

There are no test evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

Summary

Please refer to the test assurance activities for FMT_MOF.1 and FMT_SMF.1.

2.2.4 Protection of the TSF (FPT)

2.2.4.1 Firmware Update Authentication (FPT_FUA_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEEECPP-FPT_FUA_EXT.1-ASE-01

The evaluator shall examine the TSS to ensure that it describes how the TOE uses the RTU, what type of key or hash value, and where the value is stored on the RTU. The evaluator shall also verify that the TSS contains a description (storage location) of where the original firmware exists.

Summary

Section 7.1.4.2 of the [ST] describes the Apple Root CA public key, which serves as the Root of Trust for Update (RTU), its storage location, as well as the key and hash values used to verify updates. This section explains that the updates are provided by the Apple Update Server.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.2.4.2 Protection of Key and Key Material (FPT_KYP_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_KYP_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FPT_KYP_EXT.1-ASE-01

[TD0458]

The evaluator shall examine the TSS and verify it identifies the methods used to protect keys stored in non-volatile memory.

Summary

Section 7.1.4.1 of the [ST] states that all symmetric keys are cryptographically wrapped before being stored in non-volatile memory.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_KYP_EXT.1-AKM-01

Assurance Activity AA-FDEEECPP-FPT_KYP_EXT.1-AKM-01

[TD0458]

The evaluator shall verify the KMD to ensure it describes the storage location of all keys and the protection of all keys stored in non-volatile memory. The description of the key chain shall be reviewed to ensure the selected method is followed for the storage of wrapped or encrypted keys in

non-volatile memory and plaintext keys in non-volatile memory meet one of the criteria for storage.

Summary

Table 1 of the [KMD] lists each type of key, its memory type (volatile or non-volatile) and how it is protected. The evaluator verified the description of the key chain (Section 3.2 of the [KMD]) and confirmed that all wrapped keys in non-volatile memory are encrypted using AES-KW. The only plaintext key is the Secure Enclave Unique ID, which is protected by the Secure Enclave Hardware.

Test Assurance Activities

No assurance activities defined.

2.2.4.3 Power Saving States (FPT_PWR_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.1-ASE-01

The evaluator shall validate the TSS contains a list of Compliant power saving states.

Summary

Section 7.1.4.3 of the [ST] lists the only Compliant power saving state: G2(S5).

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.1-AGD-01

The evaluator shall ensure that guidance documentation contains a list of Compliant power saving states. If additional power saving states are supported, then the evaluator shall validate that the guidance documentation states how non-Compliant power states are disabled.

Summary

Section 3.2 of the [CCGUIDE] explains that only the G2(S5) state is a Compliant power saving state. The low power Sleep state supported by macOS is not Compliant and must be disabled using the instructions provided in this section.

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.1-AGD-01

[TD0460]

The evaluator shall ensure that guidance documentation contains a list of Compliant power saving states. If additional power saving states are supported, then the evaluator shall validate that the guidance documentation states how the use of non-Compliant power savings states are disabled.

Summary

Section 3.2 of the [CCGUIDE] explains that only the G2(S5) state is a Compliant power saving state. The low power Sleep state supported by macOS is not Compliant and must be disabled using the instructions provided in this section.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.1-ATE-01

The evaluator shall confirm that for each listed compliant state all key/key materials are removed from volatile memory by using the test defined in FCS_CKM.4(d).

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE, causing it to enter the Compliant power saving state. Then, the evaluator did not enter any password and verified that the keys were not loaded. This implies that all keys and key material is removed from volatile memory in the Compliant power saving state.

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.1-ATE-01

The evaluator shall confirm that for each listed Compliant state all key/key materials are removed from volatile memory by using the test indicated by the selection in FCS_CKM_EXT.6.

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE, causing it to enter the Compliant power saving state. Then, the evaluator did not enter any password and verified that the keys were not loaded. This implies that all keys and key material is removed from volatile memory in the Compliant power saving state.

2.2.4.4 Timing of Power Saving States (FPT_PWR_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.2-ASE-01

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.2-ASE-01

The evaluator shall validate that the TSS contains a list of conditions under which the TOE enters a Compliant power saving state.

Summary

Section 7.1.4.3 of the [ST] specifies that the Compliant power saving state is entered when the device is powered down.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.2-AGD-01

The evaluator shall check that the guidance contains a list of conditions under which the TOE enters a Compliant power saving state. Additionally, the evaluator shall verify that the guidance documentation states whether unexpected power-loss events may result in entry to a non-Compliant power saving state and, if that is the case, validate that the documentation contains information on mitigation measures.

Summary

Section 3.2 of the [CCGUIDE] lists the conditions under which the TOE enters the Compliant power saving state (the device powering down). Additionally, it states that an unexpected power-loss event causes the TOE to power down, thereby entering the Compliant power saving state.

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.2-AGD-01

The evaluator shall check that the guidance contains a list of conditions under which the TOE enters a Compliant power saving state. Additionally, the evaluator shall verify that the guidance documentation provides information on how long it is expected to take for the TOE to fully transition into the Compliant power saving state (e.g. how many seconds for the volatile memory to be completely cleared).

Summary

Section 3.2 of the [CCGUIDE] lists the conditions under which the TOE enters the Compliant power saving state (the device powering down). Additionally, it describes a time frame in seconds regarding how long it takes to transition into the Compliant power saving state.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_PWR_EXT.2-ATE-01

The evaluator shall trigger each condition in the list of identified conditions and ensure the TOE ends up in a compliant power saving state by running the test identified in FCS_CKM.4(d).

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE, causing it to enter the Compliant power saving state. Then, the evaluator did not enter any password and verified that the keys were not loaded. This implies that all keys and key material is removed from volatile memory in the Compliant power saving state.

Assurance Activity AA-FDEEECPP-FPT_PWR_EXT.2-ATE-01

The evaluator shall trigger each condition in the list of identified conditions and ensure the TOE ends up in a Compliant power saving state by using the test indicated by the selection in FCS_CKM_EXT.6.

Summary

The evaluator used a tool provided by the vendor to read the key encryption keys and wrapped data encryption keys loaded by the Secure Enclave. The evaluator first rebooted the TOE, causing it to enter the Compliant power saving state. Then, the evaluator did not enter any password and verified that the keys were not loaded. This implies that all keys and key material is removed from volatile memory in the Compliant power saving state.

2.2.4.5 TSF Testing (FPT_TST_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_TST_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FPT_TST_EXT.1-ASE-01

The evaluator shall verify that the TSS describes the known-answer self-tests for cryptographic functions.

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TOE and the method by which the TOE tests those functions. The evaluator shall verify that the TSS includes each of these functions, the method by which the TOE verifies the correct operation of the function. The evaluator shall verify that the TSF data are

appropriate for TSF Testing. For example, more than blocks are tested for AES in CBC mode, output of AES in GCM mode is tested without truncation, or 512-bit key is used for testing HMAC-SHA-512.

If FCS_RBG_EXT.1 is implemented by the TOE and according to NIST SP 800-90, the evaluator shall verify that the TSS describes health tests that are consistent with section 11.3 of NIST SP 800-90.

If any FCS_COP functions are implemented by the TOE, the TSS shall describe the known-answer self-tests for those functions.

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe, for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall determine that all of the identified functions/components are adequately tested on start-up.

Summary

Section 7.1.4.4 of the [ST] provides a description of the self-tests used by the TOE. The evaluator verified that the TOE implements known answer tests for the implemented cryptographic functions, specifically:

- CTR_DRBG: health tests as specified in Section 11.3 of SP 800-90Ar1
- Other algorithms: known answer tests

The same section also describes the secure boot process ensuring that all software executing on the TOE is cryptographically signed by Apple.

The evaluator confirmed that these tests are executed on start-up.

Guidance Assurance Activities

No assurance activities defined.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.2.4.6 Trusted Update (FPT_TUD_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_TUD_EXT.1-ASE-01

Assurance Activity AA-FDEEECPP-FPT_TUD_EXT.1-ASE-01

The evaluator shall examine the TSS to ensure that it describes information stating that an authorized source signs TOE updates and will have an associated digital signature. The evaluator shall examine the TSS contains a definition of an authorized source along with a description of how the TOE uses public keys for the update verification mechanism in the Operational Environment. The evaluator ensures the TSS contains details on the protection and maintenance of the TOE update credentials.

If the Operational Environment performs the signature verification, then the evaluator shall examine the TSS to ensure it describes, for each platform identified in the ST, the interface(s) used by the TOE to invoke this cryptographic functionality.

Summary

Section 7.1.4.2 of the [ST] contains information regarding the authorized source used to sign TOE updates (Apple Update Server). Moreover, the TSS explains how the Apple Root CA

public key, stored in Boot ROM, is used to verify the update. The TOE does not use the operational environment to perform signature verification.

Guidance Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_TUD_EXT.1-AGD-01

Assurance Activity AA-FDEEECPP-FPT_TUD_EXT.1-AGD-01

The evaluator ensures that the operational guidance describes how the TOE obtains vendor updates to the TOE; the processing associated with verifying the digital signature of the updates (as defined in FCS_COP.1(a)); and the actions that take place for successful and unsuccessful cases.

Summary

Section 2.2 of the [CCGUIDE] provides instruction for obtaining vendor updates to the TOE. It also describes how the TOE performs signature verification of each update before it is installed. In the event of failed signature verification, the update is aborted.

Key Management Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDEAACPP-FPT_TUD_EXT.1-ATE-01

Assurance Activity AA-FDEEECPP-FPT_TUD_EXT.1-ATE-01

The evaluators shall perform the following tests (if the TOE supports multiple signatures, each using a different hash algorithm, then the evaluator performs tests for different combinations of authentic and unauthentic digital signatures and hashes, as well as for digital signature alone):

Test 1: The evaluator performs the version verification activity to determine the current version of the TOE. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.

Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that an update successfully installs on the TOE. The evaluator shall perform a subset of other evaluation activity tests to demonstrate that the update functions as expected.

Summary

Test 1: The evaluator verified that the TOE has a version verification capability. The currently installed version is verified to be macOS 26.3 (25D125).

Test 2: The evaluator verified that the TOE allows for the update of the macOS version.

2.3 Security Assurance Requirements

2.3.1 Development (ADV)

2.3.1.1 Basic functional specification (ADV_FSP.1)

Assurance Activity AA-FDEAACPP-ADV_FSP.1-ADV-01

Assurance Activity AA-FDEEECPP-ADV_FSP.1-ADV-01

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

Summary

The evaluator examined the *Apple macOS 26 Tahoe: FileVault FSP Mapping* and verified it describes the purpose and method of use for each interface associated with each security functional requirement.

Assurance Activity AA-FDEAACPP-ADV_FSP.1-ADV-02

Assurance Activity AA-FDEEECPP-ADV_FSP.1-ADV-02

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

Summary

The evaluator examined the *Apple macOS 26 Tahoe: FileVault FSP Mapping* and verified it identifies and describes the parameters for each interface associated with each security functional requirement.

Assurance Activity AA-FDEAACPP-ADV_FSP.1-ADV-03

Assurance Activity AA-FDEEECPP-ADV_FSP.1-ADV-03

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

Summary

The evaluator examined the *Apple macOS 26 Tahoe: FileVault FSP Mapping* and verified it traces the interfaces to the SFRs claimed by the [ST].

2.3.2 Guidance documents (AGD)

2.3.2.1 Operational user guidance (AGD_OPE.1)

Assurance Activity AA-FDEAACPP-AGD_OPE.1-AGD-01

The evaluator shall check the requirements below are met by the operational guidance.

Operational guidance documentation shall be distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Operational guidance must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The contents of the operational guidance will be verified by the Evaluation Activities defined below

and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

- *The operational guidance shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.*
- *The TOE will likely contain security functionality that does not fall under the scope of evaluation under this cPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.*

Summary

The evaluator verified that the operational guidance, *Apple macOS 26 Tahoe: FileVault Common Criteria Guide* [CCGUIDE], is publicly distributed to administrators and users. This document is available for all platforms claimed in the [ST] and contains all information required to establish and maintain the evaluated configuration.

The evaluator used the operational guidance to assess the SFR-related assurance activities. Additionally, the evaluator verified that:

- Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.
- Section 1.2 of the [CCGUIDE] clearly describes the scope of the evaluation and which security functionality is covered.

Assurance Activity AA-FDEEECPP-AGD_OPE.1-AGD-01

The evaluator shall check the requirements below are met by the operational guidance. It should be noted that operational guidance may take the form of an "integrator's guide", where the TOE developer provides a description of the interface (e.g., commands that the Host Platform may invoke to configure a SED).

Operational guidance documentation shall be distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Operational guidance must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The contents of the operational guidance will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

- *The operational guidance shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.*
- *The operational guidance shall describe how to configure the IT environments that are supported to shut down after an administratively defined period of inactivity.*
- *The operational guidance shall identify system "sleeping" states for all supported operating environments and for each environment, provide administrative guidance on how to disable the sleep state. As stated above, the TOE developer may be providing an integrator's guide and "power states" may be an abstraction that SEDs provide at various levels – e.g., may simply provide a command that the Host Platform issues to manage the state of the device, and the Host Platform is responsible for providing a more sophisticated power management scheme.*

- *The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.*

Summary

The evaluator verified that the operational guidance, *Apple macOS 26 Tahoe: FileVault Common Criteria Guide* [CCGUIDE], is publicly distributed to administrators and users. This document is available for all platforms claimed in the [ST] and contains all information required to establish and maintain the evaluated configuration.

The evaluator used the operational guidance to assess the SFR-related assurance activities. Additionally, the evaluator verified that:

- Section 1.1 of the [CCGUIDE] states that no configuration of cryptographic engines, algorithms, or key sizes is necessary or available.
- Section 3.2 of the [CCGUIDE] provides instructions to configure the non-compliant sleep state supported by macOS. This state must be disabled using the instructions provided in this section.
- Section 3.2 of the [CCGUIDE] provides instructions to configure the non-compliant sleep state supported by macOS. This state must be disabled using the instructions provided in this section.
- Section 1.2 of the [CCGUIDE] clearly describes the scope of the evaluation and which security functionality is covered.

2.3.2.2 Preparative procedures (AGD_PRE.1)

Assurance Activity AA-FDEAACPP-AGD_PRE.1-AGD-01

Assurance Activity AA-FDEEECPP-AGD_PRE.1-AGD-01

The evaluator shall check the requirements below are met by the preparative procedures.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

Preparative procedures shall be distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

Preparative procedures must include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target). The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE itself).

Preparative procedures must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The preparative procedures must include

- *instructions to successfully install the TSF in each Operational Environment; and*

- *instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and*
- *instructions to provide a protected administrative capability.*

Summary

The evaluator verified that the operational guidance, *Apple macOS 26 Tahoe: FileVault Common Criteria Guide* [CCGUIDE], is publicly distributed to administrators and users. This document is available for all platforms claimed in the [ST] and contains all information required to establish and maintain the evaluated configuration.

The evaluator used the operational guidance to assess the SFR-related assurance activities. Additionally, Sections 1.1 and 1.4 of the [CCGUIDE] explain in an informal style how the security functionality is implemented by the TOE. The evaluator reviewed the explanation and found it to be easy to understand.

Section 2 of the [CCGUIDE] describes the procedure for installing or updating the TOE. The evaluator verified that these steps address all platforms claimed in the [ST]. Furthermore, Section 3 of the [CCGUIDE] describes how the security of the TOE can be managed as an administrator. In particular, this section contains instructions to create administrator and non-administrator accounts.

2.3.3 Life-cycle support (ALC)

No assurance activities defined.

2.3.4 Security target evaluation (ASE)

2.3.4.1 Conformance Claims (ASE_CCL.1)

Assurance Activity AA-FDEAACPP-ASE_CCL.1-ASE-01

Assurance Activity AA-FDEEECPP-ASE_CCL.1-ASE-01

The evaluator shall check that the statements of security problem definition in the PP and ST are identical.

Summary

The evaluator verified that the security problem definition statements in the ST are identical to the statements in CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E.

Assurance Activity AA-FDEAACPP-ASE_CCL.1-ASE-02

Assurance Activity AA-FDEEECPP-ASE_CCL.1-ASE-02

The evaluator shall check that the statements of security objectives in the PP and ST are identical.

Summary

The evaluator verified that the security objective statements in the ST are identical to the statements in CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E.

Assurance Activity AA-FDEAACPP-ASE_CCL.1-ASE-03

Assurance Activity AA-FDEEECPP-ASE_CCL.1-ASE-03

The evaluator shall check that the statements of security requirements in the ST include all the mandatory SFRs in the cPP, and all of the selection-based SFRs that are entailed by selections made in other SFRs (including any SFR iterations added in the ST). The evaluator shall check that if any other SFRs are present in the ST (apart from iterations of SFRs in the cPP) then these are taken only

from the list of optional SFRs specified in the cPP (the cPP will not necessarily include optional SFRs, but may do so). If optional SFRs from the cPP are included in the ST then the evaluator shall check that any selection-based SFRs entailed by the optional SFRs adopted are also included in the ST.

Summary

The evaluator verified that the security requirement statements in the ST include all of the mandatory SFRs in CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E as well as all of the selection-based SFRs required by selections made in other SFRs.

The evaluator verified that there are no additional SFRs in ST beyond the mandatory and selection-based SFRs in in CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E.

2.3.5 Tests (ATE)

2.3.5.1 Independent testing - conformance (ATE_IND.1)

Assurance Activity AA-FDEAACPP-ATE_IND.1-ATE-01

Assurance Activity AA-FDEEECPP-ATE_IND.1-ATE-01

The evaluator shall examine the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

Summary

The developer provided the [ST] and the [CCGUIDE], describing the TOE shall be installed and configured prior to testing. The Detailed Test Report (DTR) provides the test plan and test results for the testing performed on the TOE. The evaluator verified that the test configuration was consistent with the configuration described in the [ST].

Assurance Activity AA-FDEAACPP-ATE_IND.1-ATE-02

Assurance Activity AA-FDEEECPP-ATE_IND.1-ATE-02

The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state.

Summary

The evaluator verified that the TOE was installed properly, running a production version of the macOS 26.3 (25D125) operating system.

Assurance Activity AA-FDEAACPP-ATE_IND.1-ATE-03

Assurance Activity AA-FDEEECPP-ATE_IND.1-ATE-03

The evaluator shall prepare a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities. While it is not necessary to have one test case per test listed in an Evaluation Activity, the evaluator must show in the test plan that each applicable testing requirement in the SFR-related Evaluation Activities is covered.

The test plan identifies the platforms to be tested, and for any platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition and configuration of each platform to be tested, and any setup actions that are necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of

each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of any cryptographic engine to be used (e.g. for cryptographic protocols being evaluated).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives, and the expected results.

The test report (which could just be an updated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure, so that a fix was then installed and then a successful re-run of the test was carried out, then the report would show a "fail" result followed by a "pass" result (and the supporting details), and not just the "pass" result.

Summary

The evaluators prepared a Detailed Test Report (DTR) to specify all the tests covering the assurance activities in this evaluation. The DTR, which contains test requirements (i.e., test assurance activities), test procedures, expected test results, actual test results, along with the verdict, is provided to the validation body but is not published.

The test environment was set up according to a setup strategy that followed the evaluated configuration requirements specified in the [CCGUIDE] and [ST], supplemented by configurations required to perform testing. The testing was performed on a subset of the platforms claimed in the [ST]: only one model corresponding to each processor model was tested. The tested models are

- Mac14,2 (covering all platforms containing an M2 series processor)
- Mac15,8 (covering all platforms containing an M3 series processor)
- Mac16,8 (covering all platforms containing an M4 series processor)
- Mac17,2 (covering all platforms containing an M5 series processor)

The DTR contains a detailed rationale for the equivalency argument considered during testing.

The following tools were used for testing:

- TOE platforms running macOS 26.3 Tahoe
- Disk Utility version 22.7
- MacBook Pro (MacBookPro18,1, Mac16,7, and Mac15,10) laptops used by the vendor to interface with the TOE platforms.
- Proprietary specialized Apple USB-C serial/debugging cable
- Development-fused (dev-fused) TOE platforms running macOS 26.3 Tahoe
- Apple internal build of seputil
- Apple internal software

The machines are connected to the internal network using a wired connection.

2.3.6 Vulnerability assessment (AVA)

2.3.6.1 Vulnerability Survey (AVA_VAN.1)

Assurance Activity AA-FDEAACPP-AVA_VAN.1-AVA-01

Assurance Activity AA-FDEEECPP-AVA_VAN.1-AVA-01

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components apply to all systems claimed in the ST, and should identify at a minimum the processors used by the TOE. Software components include any libraries used by the TOE, such as cryptographic libraries. This additional documentation is merely a list of the name and version number of the components, and will be used by the evaluators in formulating hypotheses during their analysis.

The evaluator shall examine the documentation outlined below provided by the vendor to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

In addition to the activities specified by the CEM in accordance with Table 2 in [CPP_FDE_AA_V2.0E-SD], the evaluator shall perform the following activities.

Summary

The evaluator examined the [ST] and found that Section 1.4, Section 1.5, and Appendix A, contain the versions of the TOE's software and cryptographic libraries (Apple macOS, corecrypto cryptographic modules) and hardware (application processor, Secure Enclave, DMA Storage Controller). The evaluator's assessments cover all the platforms specified in the [ST].

Assurance Activity AA-FDEAACPP-AVA_VAN.1-AVA-02

Assurance Activity AA-FDEEECPP-AVA_VAN.1-AVA-02

The evaluator formulates hypotheses in accordance with process defined in Appendix A.1 of [CPP_FDE_AA_V2.0E-SD]. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3 of [CPP_FDE_AA_V2.0E-SD]. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2 of [CPP_FDE_AA_V2.0E-SD]. The results of the analysis shall be documented in the report according to Appendix A.3 of [CPP_FDE_AA_V2.0E-SD].

Summary

As required by NIAP Policy 17 Addendum – vulnerability mitigation guidance, the following search terms were used during the vulnerability search:

- FileVault
- macOS 26
- Secure Enclave Processor
- corecrypto
- AES-XTS
- single overwrite zeroization
- Apple M2
- Apple M3
- Apple M4
- Apple M5

The evaluator searched for publicly known vulnerabilities using the following sources:

- MITRE Common Vulnerabilities and Exposures (CVE) List:
  - https://cve.mitre.org/cve/
- National Vulnerability Database (NVD):

- https://nvd.nist.gov/

- CISA Known Exploited Vulnerabilities (KEV) Catalog:
  - https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Apple security releases
  - https://support.apple.com/en-us/100100
- Background Security Improvements by date
  - https://support.apple.com/en-us/111333

The vulnerability search was repeated on the following dates, throughout the evaluation process (last search date: 2026-07-28):

- 2026-06-29
- 2026-07-28

All “crucial” vulnerabilities (as defined by the NIAP Policy 17 Addendum) found during the vulnerability search were mitigated. Any residual vulnerabilities are not considered crucial.

Finally, the evaluators examined the Type 1 and Type 2 flaw hypotheses specified in Appendix A.1 of [CPP_FDE_AA_V2.0E-SD] and [CPP_FDE_EE_V2.0E-SD]. More information, including the process of the vulnerability analysis, is provided in the proprietary Evaluation Technical Report (ETR).

A References

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	CC:3.1 R5
	Date	April 2017
	Location	https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
	Location	https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
	Location	https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf
CCEVS-LG	**CCEVS LabGrams**	
	Author(s)	NIAP
	Date	July 2026
	Location	https://www.niap-ccevs.org/labgrams
CCEVS-PL	**CCEVS Scheme Policy Letters**	
	Author(s)	NIAP
	Date	July 2026
	Location	https://www.niap-ccevs.org/policies
CCEVS-PUB	**CCEVS Scheme Publications**	
	Author(s)	NIAP
	Date	July 2026
	Location	https://www.niap-ccevs.org/publications
CCEVS-TD	**Technical Decisions**	
	Author(s)	NIAP
	Date	July 2026
	Location	https://www.niap-ccevs.org/technical-decisions
CCGUIDE	**Apple macOS 26 Tahoe: FileVault Common Criteria Guide**	
	Date	2026-06-29
	File name	st-vid11696-agd.pdf
CEM	**Common Methodology for Information Technology Security Evaluation**	
	Version	CC:3.1 R5
	Date	April 2017
	Location	https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf
CP-P_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
CP-P_FDE_AA_V2.0E-SD	**Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition**	
	Version	2.0 + Errata 20190201
	Date	2019-02-01
	Location	https://www.niap-ccevs.org/protectionprofiles/437
CP-P_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
CP-P_FDE_EE_V2.0E-SD	**Supporting Document Mandatory Technical Document Full Drive Encryption: Encryption Engine**	
	Version	2.0 + Errata 20190201
	Date	2019-02-01
	Location	https://www.niap-ccevs.org/protectionprofiles/438
KMD	**Apple macOS 26 Tahoe: FileVault Key Management Description**	
	Version	1.0
	Date	2026-06-29
ST	**Apple macOS 26 Tahoe: FileVault Security Target**	
	Date	2026-06-29
	File name	st-vid11696-st.pdf

B Glossary

Augmentation

The addition of one or more requirement(s) to a package.

Authentication data

Information used to verify the claimed identity of a user.

Authorised user

A user who may, in accordance with the SFRs, perform an operation.

Class

A grouping of CC families that share a common focus.

Component

The smallest selectable set of elements on which requirements may be based.

Connectivity

The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

Dependency

A relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package.

Deterministic RNG (DRNG)

An RNG that produces random numbers by applying a deterministic algorithm to a randomly selected seed and, possibly, on additional external inputs.

Element

An indivisible statement of security need.

Entropy

The entropy of a random variable X is a mathematical measure of the amount of information gained by an observation of X.

Evaluation

Assessment of a PP, an ST or a TOE, against defined criteria.

Evaluation Assurance Level (EAL)

An assurance package, consisting of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale.

Evaluation authority

A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.

Evaluation scheme

The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.

Exact conformance

A subset of Strict Conformance as defined by the CC, is defined as the ST containing all of the requirements in the Security Requirements section of the PP, and potentially requirements from Appendices of the PP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in the Security Requirements section of the PP are allowed to be omitted.

Extension

The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

External entity

Any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

Family

A grouping of components that share a similar goal but may differ in emphasis or rigour.

Formal

Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Guidance documentation

Documentation that describes the delivery, preparation, operation, management and/or use of the TOE.

Identity

A representation (e.g. a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.

Informal

Expressed in natural language.

Object

A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

Operation (on a component of the CC)

Modifying or repeating that component. Allowed operations on components are assignment, iteration, refinement and selection.

Operation (on an object)

A specific type of action performed by a subject on an object.

Operational environment

The environment in which the TOE is operated.

Organisational Security Policy (OSP)

A set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment.

Package

A named set of either functional or assurance requirements (e.g. EAL 3).

PP evaluation

Assessment of a PP against defined criteria.

Protection Profile (PP)

An implementation-independent statement of security needs for a TOE type.

Random number generator (RNG)

A group of components or an algorithm that outputs sequences of discrete values (usually represented as bit strings).

Refinement

The addition of details to a component.

Role

A predefined set of rules establishing the allowed interactions between a user and the TOE.

Secret

Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.

Secure state

A state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

Security attribute

A property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

Security Function Policy (SFP)

A set of rules describing specific security behaviour enforced by the TSF and expressible as a set of SFRs.

Security objective

A statement of intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.

Security Target (ST)

An implementation-dependent statement of security needs for a specific identified TOE.

Seed

Value used to initialize the internal state of an RNG.

Selection

The specification of one or more items from a list in a component.

Semiformal

Expressed in a restricted syntax language with defined semantics.

ST evaluation

Assessment of an ST against defined criteria.

Subject

An active entity in the TOE that performs operations on objects.

Target of Evaluation (TOE)

A set of software, firmware and/or hardware possibly accompanied by guidance.

TOE evaluation

Assessment of a TOE against defined criteria.

TOE resource

Anything useable or consumable in the TOE.

TOE Security Functionality (TSF)

A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs.

Transfers outside of the TOE

TSF mediated communication of data to entities not under control of the TSF.

True RNG (TRNG)

A device or mechanism for which the output values depend on some unpredictable source (noise source, entropy source) that produces entropy.

Trusted channel

A means by which a TSF and a remote trusted IT product can communicate with necessary confidence.

Trusted path

A means by which a user and a TSF can communicate with necessary confidence.

TSF data

Data created by and for the TOE, that might affect the operation of the TOE.

TSF Interface (TSFI)

A means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF.

User

See external entity.

User data

Data created by and for the user, that does not affect the operation of the TSF.