KLC Group LLC

CipherDriveOne Kryptr 1.1.0

Assurance Activity Report

Version 0.3

April 2024

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Reviewer	Description
0.1	2023-04-19	S. Rae		TDs Applied, AAR updated with all relevant cPPs, initial TSS draft
0.2	2024-03-26	K. Steiner	C. Cantlon	TSS activities; finalizing; release to QA
0.3	2024-04-23	K. Steiner	C. Cantlon	Addressing ECR comments

Table of Contents

1 Introduction

1 This Assurance Activity Report (AAR) documents the evaluation activities performed by Lightship Security USA for the evaluation identified in Table 1. The AAR is produced in accordance with National Information Assurance Program (NIAP) reporting guidelines.

1.1 Evaluation Identifiers

Table 1: Evaluation Identifiers

Scheme	NIAP
Evaluation Facility	Lightship Security USA 3600 O'Donnell St., Suite 2 Baltimore, MD 21224
Developer/Sponsor	KLC Group LLC
TOE	KLC Group LLC CipherDriveOne Kryptr 1.1.0
Security Target	KLC Group LLC CipherDriveOne Kryptr 1.1.0 Security Target, Version 1.5
Protection Profile	• collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0 + Errata 20190201 • collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0 + Errata 20190201

1.2 Evaluation Methods

2 The evaluation was performed using the methods, tools and standards identified in Table 2.

Table 2: Evaluation Methods

Evaluation Criteria	CC v3.1 R5
Evaluation Methodology	CEM v3.1R5
Supporting Documents	• Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition, February 2019, Version 2.0 + Errata 20190201 • Supporting Document Mandatory Technical Document Full Drive Encryption: Encryption Engine, February 2019, Version 2.0 + Errata 20190201
Tools	A description of tools is provided in [DTR].

Table 3: Technical Decisions

TD #	Name	Applicability Rationale	Source
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Applicable	CPP_FDE_AA CPP_FDE_EE
TD0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Applicable	CPP_FDE_EE
TD0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Applicable	CPP_FDE_EE
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	Not Applicable - TOE is not a NAS	CPP_FDE_AA CPP_FDE_EE
TD0759	FIT Technical Decision for FCS_AFA_EXT.1.1	Applicable	CPP_FDE_AA
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	Applicable	CPP_FDE_AA
TD0764	FIT Technical Decision for FCS_PCC_EXT.1	Applicable	CPP_FDE_AA
TD0765	FIT Technical Decision for FMT_MOF.1	Applicable	CPP_FDE_AA
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	Applicable	CPP_FDE_AA CPP_FDE_EE
TD0767	FIT Technical Decision for FMT_SMF.1.1	Applicable	CPP_FDE_AA
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	Applicable	CPP_FDE_AA CPP_FDE_EE

1.3 Summary of SFRs

Table 4: List of SFRs

Requirement	Title
AA:FCS_AFA_EXT.1	Authorization Factor Acquisition
AA:FCS_AFA_EXT.2	Timing of Authorization Factor Acquisition
AA:FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
AA:FCS_CKM.4(d)	Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
AA:FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)

Requirement	Title
AA:FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
AA:FCS_KYC_EXT.1	Key Chaining (Initiator)
AA:FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
AA:FMT_MOF.1	Management of Functions Behavior
AA:FMT_SMF.1	Specification of Management Functions
AA:FMT_SMR.1	Security Roles
AA:FPT_KYP_EXT.1	Protection of Key and Key Material
AA:FPT_PWR_EXT.1	Power Saving States
AA:FPT_PWR_EXT.2	Timing of Power Saving States
AA:FPT_TUD_EXT.1	Trusted Update
EE:FCS_CKM.1(c)	Cryptographic Key Generation (Data Encryption Key)
EE:FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
EE:FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
EE:FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
EE:FCS_CKM_EXT.6	Cryptographic Key Destruction Types
EE:FCS_KYC_EXT.2	Key Chaining (Recipient)
EE:FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
EE:FCS_VAL_EXT.1	Validation
EE:FDP_DSK_EXT.1	Protection of Data on Disk
EE:FMT_SMF.1	Specification of Management Functions
EE:FPT_KYP_EXT.1	Protection of Key and Key Material
EE:FPT_PWR_EXT.1	Power Saving States
EE:FPT_PWR_EXT.2	Timing of Power Saving States
EE:FPT_TST_EXT.1	TSF Testing
EE:FPT_TUD_EXT.1	Trusted Update

Requirement	Title
Selection based	
AA:FCS_CKM.1(b)	Cryptographic Key Generation (Symmetric Keys)
AA/EE:FCS_COP.1(a)	Cryptographic Operation (Signature Verification)
AA:FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
AA:FCS_COP.1(c)	Cryptographic Operation (Keyed Hash Algorithm)
AA:FCS_COP.1(g)	Cryptographic Operation (Key Encryption)
AA:FCS_KDF_EXT.1	Cryptographic Key Derivation
AA:FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning
AA:FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)
AA:FCS_SMC_EXT.1	Submask Combining
EE:FCS_CKM.4(d)	Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
EE:FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
EE:FCS_COP.1(c)	Cryptographic Operation (Keyed Hash Algorithm)
EE:FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption)
EE:FCS_COP.1(g)	Cryptographic Operation (Key Encryption)
EE:FCS_KDF_EXT.1	Cryptographic Key Derivation
EE:FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)

1.4 Reference Documents

Table 5: List of Reference Documents

Ref	Document
[ST]	KLC Group LLC CipherDriveOne Kryptr 1.1.0 Security Target, Version 1.5, April 2024
[KMD]	KLC Group CipherDriveOne Kryptr 1.1.0 Key Management Description, Version 1.4, April 2024
[AGD]	KLC Group LLC CipherDriveOne Kryptr 1.1.0 Common Criteria Guide, Version 1.1, April 2024
[MAN]	KLC Group LLC CipherDriveOne Kryptr Administrator Guide V 1.0.1, 4-18-2024

Ref	Document
[DTR]	KLC Group LLC CipherDriveOne Kryptr 1.1.0 cPP FDE AA+EE 2.0E Test Plan, Version 0.4, April 2024
[EVIDENCE]	KLC Group LLC CIpherDriveOne Kryptr 1.1.0 cPP FDE AA+EE 2.0E Test Evidence, Version 0.4, April 2024
[VULN]	KLC Group LLC CipherDriveOne Kryptr 1.1.0 Vulnerability Assessment, Version 0.2, April 2024
[ETR]	KLC Group CipherDriveOne Kryptr 1.1.0 Evaluation Technical Report, Version 0.3, April 2024
[PP-AA]	collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, Version 2.0 + Errata 20190201
[PP-EE]	collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201
[SD-AA]	Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition, February 2019, Version 2.0 + Errata 20190201
[SD-EE]	Supporting Document Mandatory Technical Document Full Drive Encryption: Encryption Engine, February 2019, Version 2.0 + Errata 20190201

2 Test Overview

3 Testing was performed by Nathan Bennett, Nil Folquer, and Kevin Steiner from September 2023 through April 2024.

4 All testing performed in the Lightship Baltimore facility that has been accredited by NVLAP. The TOE and test setup was physically and logically protected from unauthorized access, so the integrity of the TOE and test results can be assured.

2.1 TOE Models/Platforms

5 The physical boundary of the TOE encompasses the KLC software (including Linux Kernel 5.15). Users download the software after purchase from KLC's web portal.

6 The TOE (KLC Group LLC CipherDriveOne Kryptr 1.1.0, build 17) was installed with the following non-TOE components:

a) **Protected OS.** The TOE supports protection of the following Linux Operating Systems and Windows Operating Systems:

   i) Red Hat Enterprise Linux 8

   ii) Red Hat Enterprise Linux 9

   iii) Microsoft Windows 10

   iv) Microsoft Windows 11

   CC Testing was performed using the following operating systems:

   - Red Hat Enterprise Linux 9
   - Microsoft Windows 11

b) **Computer Hardware.** 64-bit Intel-based UEFI booted systems that supports Intel Secure Key Technology. CC testing was performed using the following CPU:

   1. Intel Core i7-1265U (Alder Lake)

c) **Smartcard and reader.** When dual factor authentication is used, Federal Information Processing Standard (FIPS) 201 Personal Identity Verification Common Access Card (PIV-CAC) compliant smartcards and readers are required.

2.1.1 Test Platform Equivalency

7 KLC Group LLC CipherDriveOne Kryptr 1.1.0 was fully tested with both Red Hat Enterprise Linux 9 and Microsoft Windows 11 platforms. Equivalency rationale for the [PP-AA] and [PP-EE] equivalency considerations is provided in detail in [DTR].

3 Evaluation Activities for SFRs (CPP_FDE_AA_V2.0E)

3.1 Cryptographic Support (FCS)

3.1.1 NIAP Policy 5

8 To demonstrate that all cryptographic requirements are satisfied, the Assurance Activity Report must clearly indicate all SFRs for which a CAVP certificate is claimed and include, at a minimum, the cryptographic operation, the NIST standard, the SFR supported, the CAVP algorithm list name (e.g. AES, KAS, CVL, etc.) and the CAVP Certificate number.

SFR	Cryptographic Operation	NIST/ISO Standard	CAVP Certificate (Algorithm)
AA/EE:FCS_COP.1(a)	RSA Digital Signature Algorithm with a key size (modulus) of [3072-bit] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 29 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA schemes.	FIPS PUB 186-4	A5166 (RSA)
AA:FCS_COP.1(b)	cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-384.	ISO/IEC 10118-3:2004	A5166 (SHA-384)
AA:FCS_COP.1(c)	cryptographic keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-384 and cryptographic key size 384 bits.	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	A5166 (HMAC-SHA-384)
AA:FCS_COP.1(g)	key encryption and decryption in accordance with a specified cryptographic algorithm AES used in CBC mode and cryptographic key sizes 128 bits, 256 bits.	AES as specified in ISO /IEC 18033-3, CBC as specified in ISO/IEC 10116	A5166 (AES-CBC (128,256))
AA:FCS_RBG_EXT.1	random bit generation services using CTR_DRBG (AES)	NIST SP 800-90A	A5166 (CTR_DRBG (AES-256))

SFR	Cryptographic Operation	NIST/ISO Standard	CAVP Certificate (Algorithm)
EE:FCS_COP.1(b)	cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-256, SHA-384.	ISO/IEC 10118-3:2004	A3241 (SHA-384) A3230 (SHA-256, SHA-384) A3231 (SHA-384)
EE:FCS_COP.1(c)	cryptographic message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-256, HMAC-SHA-384 and cryptographic key sizes 256, 384 bits used in HMAC.	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	A3241 (HMAC-SHA-384) A3230 (HMAC-SHA-256, HMAC-SHA-384) A3231 (HMAC-SHA-384)
EE:FCS_COP.1(f)	data encryption and decryption in accordance with a specified cryptographic algorithm AES used in XTS mode and cryptographic key sizes 128 bits, 256 bits.	AES as specified in ISO /IEC 18033-3, XTS as specified in IEEE 1619	A3241 (AES-XTS (128,256)) A3230 (AES-XTS (128,256)) A3231 (AES-XTS (128,256))
EE:FCS_COP.1(g)	key encryption and decryption in accordance with a specified cryptographic algorithm AES used in CBC mode and cryptographic key sizes 128 bits, 256 bits.	AES as specified in ISO /IEC 18033-3, [CBC as specified in ISO/IEC 10116	A3241 (AES-CBC (128,256)) A3230 (AES-CBC (128,256)) A3231 (AES-CBC (128,256))
EE:FCS_RBG_EXT.1	random bit generation services HMAC_DRBG (SHA-256)	NIST SP 800-90A	A3230 (HMAC-DRBG (SHA-256))

3.1.2 AA: FCS_AFA_EXT.1 Authorization Factor Acquisition

3.1.2.1 TSS

9 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.

Findings:	[ST] Section 6.2.1 states that the authentication factors for the TOE are username/Password and smartcards (dual-factor). Section 2.3 of the [AGD] lists all authentication factors, provides a description of them, and provides references to instruction for modification in [MAN].

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

Findings:	[ST] Sections 6.2.1.1 and 6.2.1.2 describe how the authorization factors are input to the TOE for both username/password and dual-factor authentication.

3.1.2.2 Operational Guidance

11 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.

Findings:	Section 2.3 of the [AGD] is dedicated to identifying the authorization factors for the TOE. These factors include password and multi-factor using username, password and smart card. This section provides a short description of each. Additional instructions for the set up of each of these authentication factors can be referenced in the [MAN] under the following sections: Password: “Add a Password User” Multi-Factor: “Add a MFA (Multifactor Authentication) User”

3.1.2.3 KMD

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

Findings:	[KMD] Section 3.3 provides Figure 9: Authentication / Unlock which demonstrates how the initial authorization factor(s) (username/password or dual-factor) directly contributes to unwrapping the BEV.

13 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).

Findings:	[KMD] Section 3.4.2 describes how each of the submasks are produced from the authorization factor using PBKDF2. Section 3.4.4 describes the use of PBKDF2 in detail, including the options, output key length and standard to which is conforms.

3.1.2.4 Test

14 The password authorization factor is tested in FCS_PCC_EXT.1.

15 The evaluator shall also perform the following tests:

16 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.

High-Level Test Description
The evaluator attempted to log into the TOE independently with no password, correct password and no smartcard, and no password and correct smartcard and observed in each case that access to decrypted plaintext data was not granted.
Findings: PASS

3.1.3 AA: FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

3.1.3.1 TSS

17 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.

Findings:	[ST] Section 6.2.2 states that the user must authenticate via password or dual factor in order to gain access to user data after the TOE enters a compliant power saving state. The power saving states are further detailed by AA:FPT_PWR_EXT.1. The chosen authorization factors of password and smartcard (dual-factor) are consistent with selections made for AA:FCS_AFA_EXT.1.1.

3.1.3.2 Operational Guidance

18 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.

Findings:	[AGD] Section 2.5 describes the power saving states and states “When resuming from the above power saving states, users are required to re-authenticate using the same authorization factors as per normal operation.”

3.1.3.3 KMD

19 There are no KMD evaluation activities for this SFR.

3.1.3.4 Test

20 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.

High-Level Test Description
The evaluator successfully logged into the TOE and was granted access to decrypted plaintext data. Then for each power state claimed, the evaluator tried to log in using an incorrect password, using an incorrect password with a correct smartcard and a correct password with an incorrect

High-Level Test Description
smartcard and observed that in each attempt the TOE did not grant access to decrypted plaintext data.
Findings: PASS

3.1.4 AA: FCS_CKM.4(a) Cryptographic Key Destruction (Power Management)

3.1.4.1 TSS

21 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.

Findings:	Section 6.2.4 of the [ST] states that the TOE erases cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state per FPT_PWR_EXT.1 using the key destruction method specified in AA:FCS_CKM.4(d). Temporary keys are tracked by allocated memory throughout their usage lifecycle until destruction. The OE does not destroy keys from volatile memory for this activity thus requires no interface details.

3.1.4.2 Operational Guidance

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

Findings:	[AGD] Section 2.4 states that when a user initiates a request to enter a power saving state, the TOE will instruct the protected OS to destroy all cryptographic keys and key material from volatile memory.

3.1.4.3 KMD

23 The evaluator shall check to ensure the KMD lists each type of key, its origin, possible memory locations in volatile memory.

Findings:	Table 3 of the [KMD] lists all keys, their origin and storage location.

3.1.4.4 Test

24 There are no test evaluation activities for this SFR.

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

3.1.5.1 TSS + KMD (Key Management Description may be used if necessary details describe proprietary information)

25 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.

Findings:	[KMD] is used to satisfy this activity. Section 3.1 provides Table 3: Key Lifecycle which lists all keys stored in volatile memory, how they are derived and/or wrapped, and their destruction method. For each key in volatile memory, this table describes how they are overwritten. The evaluator confirmed that this description is consistent with the AA:FCS_CKM.4(d) claims.

26 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).

Findings:	[KMD] is used to satisfy this activity. Section 3.1 provides Table 3: Key Lifecycle which lists all keys stored in non-volatile memory, how they are derived and/or wrapped, and their destruction method. For each key in non-volatile memory, this table describes how they are destroyed. The evaluator confirmed that this description is consistent with the AA:FCS_CKM.4(d) claims.

27 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.

Findings:	[KMD] is used to satisfy this activity. Section 3.1 provides Table 3: Key Lifecycle for all keys. For each media type, the storage location and destruction method is consistent with the AA:FCS_CKM.4(d) selections and the TSS.

28 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.

Findings:	No circumstances are identified in which the TOE does not strictly conform to the key destruction requirement. [ST] does not make use of the open assignment.

3.1.5.2 Operational Guidance

29 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.

30 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.

31 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.

32 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.

33 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.

Findings:	Section 2.4 of the [AGD] states that the TOE handles the destruction of cryptographic keys and key material when they are no longer required. There are no situations where key destruction would be delayed or prevented. The TOE also instructs the protected OS to destroy all cryptographic keys and key material in volatile memory when entering a power saving state.

3.1.5.3 Test

Technical Decision: The evaluation activities were modified per TD0766.

34 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.

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

36 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.

High-Level Test Description
The evaluator created a new admin user in the system and set up a password and a smartcard. The evaluator then performed actions on the TOE so that the key was destroyed. The evaluator then dumped the content of the memory, searched for the key value and the key value broken into thirds inside the memory dump and confirmed it was removed from the memory. This was repeated for all keys and power states and for both 128-bit and 256-bit keychains.
Findings: PASS

37 *The following tests apply only 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.*

38 *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.*

39 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.

High-Level Test Description
The above test is not applicable since the ST selects "*instructs the underlying platform…"*.
Findings: N/A

40 *The following tests apply only 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.

41 *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.*

42 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.

43 The test succeeds if correct pattern is used to overwrite the key in the memory location.

44 If the pattern is not found the test fails.

High-Level Test Description
The above test is not applicable since the ST selects "*instructs the underlying platform…"*.
Findings: N/A

3.1.6 AA: FCS_CKM_EXT.4(a) Cryptographic Key and Key Material Destruction (Destruction Timing)

3.1.6.1 TSS

45 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.

Findings:	[ST] Section 6.2.6 states that all keys and key material are destroyed when they are no longer needed. [KMD] Section 3.1 also contains Table 3: Key Lifecycle which details the exact conditions in which each key is destroyed.

3.1.6.2 Operational Guidance

46 There are no AGD evaluation activities for this SFR.

3.1.6.3 KMD

47 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.

Findings:	Table 3 in [KMD] identifies "End-of-life / When key is destroyed" which describes when keys and key material are no longer needed for each key/CSP.

48 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.

Findings:	Table 3 in [KMD] describes areas where keys and key material reside and when the keys and key material are no longer needed, and it also includes a key lifecycle and key destruction description. The evaluator confirmed that the means by which the keys are destroyed follow AA:FCS_CKM.4(a) claims.

3.1.6.4 Test

49 There are no test evaluation activities for this SFR.

3.1.7 AA: FCS_CKM_EXT.4(b) Cryptographic Key and Key Material Destruction (Power Management)

3.1.7.1 TSS

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

Findings:	[ST] Section 6.2.7 states that when the TOE transitions into a power saving state (as defined by AA:FPT_PWR_EXT.1 that all of the following are destroyed: key material, BEV and authentication factors stored in plaintext.

3.1.7.2 Operational Guidance

51 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.

Findings:	Section 2.5 of the [AGD] lists all compliant power saving states. Entering a compliant power saving state is initiated by the user through the Host OS. Section 2.5 also states that an unexpected power loss would result in the G3 power state. Therefore, there are no instances where the TOE may enter a non-compliant power saving state.

3.1.7.3 KMD

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

Findings:	Table 3 in [KMD] contains the storage location for all keys and key material.

53 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.

Findings:	Table 3 in [KMD] contains the storage location, purpose, and end-of-life / when the key is destroyed for all keys and key material.

3.1.7.4 Test

54 There are no test evaluation activities for this SFR.

3.1.8 AA: FCS_KYC_EXT.1 Key Chaining (Initiator)

3.1.8.1 TSS

55 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.

Findings:	Section 6.2.13 of the [ST] states that the TOE supports a default BEV (AK) of 256 bits. 128 bits can also be used via licencing provided by the developer. Per section 6.2.26, when 128 bits is enabled, the TOE uses AES-XTS 128 data encryption and when the default 256 bits is used the TOE uses AES-XTS 256 data encryption. Additionally, per section 6.2.11 the TOE uses AES-CBC-128 or AES-CBC-256 for key encryption. The evaluator confirmed that regardless of the key size used, the keychain maintains the effective strength of the BEV.

3.1.8.2 Operational Guidance

56 There are no AGD evaluation activities for this SFR.

3.1.8.3 KMD

57 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.

Findings:	[KMD] Section 3.4 provides a high level description of the key hierarchy for password and dual-factor authorization methods which is consistent with AA:FCS_AFA_EXT.1. FCS_COP.1(d) is not claimed in the [ST]. AA:FCS_KDF_EXT.1 claims a conditioned submask (PBKDF2) which is present in the keychain figures in section 3.4 of the [KMD].

58 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.

Findings:	[KMD] Section 3.4 displays figures for each keychain process function, the hierarchy and what the key material is derived from. Storage location for each key can be referenced in Table 3 of the [KMD]. Regardless of the 128-bit or 256-bit BEV, the BEV strength is maintained throughout the keychain.

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

Findings:	The figures in [KMD] section 3.4 show the strength of the keychain. The first paragraph of section 3.4 also states that the keychains are designed to protect a 256 or 128-bit DEK.

3.1.8.4 Test

60 There are no test evaluation activities for this SFR.

3.1.9 AA: FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

3.1.9.1 TSS

Technical Decision: The evaluation activities were modified per TD0760.

61 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.

Findings:	Section 6.2.17 of the [ST] states that 32 byte salts are generated using RAND_bytes from the DRBG specified in AA:FCS_RBG_EXT.1.1 at the time of encryption.

62 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.

Findings:	Section 6.2.17 of the [ST] states that the TOE does not make use of nonces. IVs are generated by the DRBG using RAND_bytes and are appended to the encrypted data. Tweaks are not used by AES-CBC and not considered. The evaluator has determined further that due to the selection for AA:FCS_SNI_EXT.1.3 that IVs are non-repeating and unpredictable.

3.1.9.2 Operational Guidance

63 There are no AGD evaluation activities for this SFR.

3.1.9.3 KMD

64 There are no KMD evaluation activities for this SFR.

3.1.9.4 Test

65 There are no test evaluation activities for this SFR.

3.2 Security management (FMT)

66 The evaluator shall perform the following test for each method of local login allowed:

3.2.1 AA: FMT_MOF.1 Management of Functions Behavior

3.2.1.1 TSS

67 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.

Findings:	Section 6.4.1 of the [ST] states that the TOE does not allow any modification related to power saving states.

3.2.1.2 Operational Guidance

68 The evaluator to check if guidance documentation describes which authorization factors are required to change Compliant power saving state behavior and properties.

Findings:	Section 2.5 of the [AGD] lists all power saving states. Entering a compliant power saving state is initiated by the user through the Host OS. Section 6.4.1 of the [ST] informs that no modifications of the power states of any kind is allowed.

3.2.1.3 KMD

69 There are no KMD evaluation activities for this SFR.

3.2.1.4 Test

Technical Decision: The evaluation activities were modified per TD0765.

70 The evaluator shall perform the following tests:

71 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.

High-Level Test Description
Test 1 is not applicable since the TOE does not allow any modification related to power saving states.
Findings: N/A

72 Test 2(conditional): If the product support 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.

High-Level Test Description
Test 2 is not applicable since the TOE does not allow any modification related to power saving states.
Findings: N/A

3.2.2 AA: FMT_SMF.1 Specification of Management Functions

3.2.2.1 TSS

73 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.

Findings:	Section 6.4.2 of the [ST] states that the TOE sends the request to the EE to change the DEK in the following manner: a) **Linux:** On user's request, luksAddKey and luksKillSlot commands are sent to the CDOCryptsetup engine using the Admin AK.

	b) **Windows:** On user's request, CDO sends-ChangeDEK command to CDOCryptsetup engine using the Admin AK.

74 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.

Findings:	Section 6.4.2 of the [ST] states that the TOE sends the request to the EE to cryptographically erase the DEK in the following manner: a) **Linux** On user's request for cryptographic erase, luksAddKey command is sent to the cryptsetup engine using the Admin AK with the value of a new random key followed by luksKillSlot for Admin and User slots. b) **Windows**: On user's request for cryptographic erase, CDO sends ChangeDEK command to CDOCryptsetup engine using the Admin AK.

75 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.

Findings:	[ST] Section 6.4.2 states that the TOE GUI may be used by the user to change their password. The TOE GUI may also be used for new smartcard enrolments with a changed PIN.

76 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.

Findings:	[ST] Section 6.4.2 states that the TOE GUI (maintenance screen) can be used to initiate updates.

77 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.

Findings:	Section 6.4.1 of the [ST] states that the TOE does not allow any modification related to power saving states. AA:FMT_SMF.1 adds "configure authorization factors" and "disable key recovery functionality" for e). Section 6.4.2 of the [ST] describes how to configure authorization factors as well as how to disable key recovery functionality.

3.2.2.2 Operational Guidance

78 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.

Findings:	Section 2.6 of the [AGD] states that further instructions for the changing or erasing of DEK can be referenced in [MAN] sections labelled "Change DEK" and "Erase Disk".

79 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.

Findings:	Section 2.6 of the [AGD] states that further instructions for the changing of authorization factor can be referenced in [MAN] sections labelled “Update Password User” and “Update Smartcard User”.

80 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.

Findings:	Section 2.7 of the [AGD] provides information on how the TOE can be updated. Software updates files must have to be manually downloaded from the KLC web portal and then copied to a USB drive. The UI is then used to trigger the update from USB. Detailed Update instructions are provided in [MAN] section “CDO Kryptr Upgrade”.

81 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.

Findings:	This is N/A since the TOE does not arrive with any default authorization factors.

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

Findings:	Section 2.10 of the [AGD] describes the general capabilities around key recovery functionality. Key recovery functionality (export configuration or backup database) can be disabled at install time (using ‘-n noexport’ as one of the command-line parameters) or (if installed) recovery can be administratively disabled at runtime (by unchecking the ‘Recovery’ configuration item in the Settings Console as the Security Officer). Further instructions are provided in [MAN] section ‘*Installation of CDO Kryptr’* subsection *‘CDO Kryptr Install Optional Parameters’* subsection ‘*Install CDO Kryptr with exported configuration file’.*

83 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).

Findings:	Section 2.5 of the [AGD] list all power saving states supported by the TSF. The TOE does not allow any modification related to power saving states.

3.2.2.3 KMD

84 There are no KMD evaluation activities for this SFR.

3.2.2.4 Test

85 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.

High-Level Test Description
The evaluator exercised the Change DEK and Erase Disk commands and confirmed that the operations were performed successfully.

High-Level Test Description
Findings: PASS

86 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.

87 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.

High-Level Test Description
The evaluator logged into the TOE and changed the password of the user and attempted to log in using the previous password to verify that the TOE rejects the log in attempt and does not grant access to the decrypted data. The evaluator then enabled dual-factor authentication and confirmed that using the correct password and smartcard granted access to decrypted data. The evaluator then attempted to log into the TOE with the correct password but incorrect smartcard and confirmed that access to decrypted data was not granted by the TOE.
Findings: PASS

88 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.

High-Level Test Description
This was performed in conjunction with FPT_TUD_EXT.1.
Findings: PASS

89 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.

High-Level Test Description
The ST selects 'disable key recovery functionality' and 'configure authorization factors'. The 'disable key recovery functionality' selection is covered in AA:FMT_SMF.1 Test 3 below. The 'configure authorization factors' selection is covered in the AA:FMT_SMF.1 item c) testing above.
Findings: PASS

Technical Decision: This Test was modified per TD0767.

90 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.

High-Level Test Description
This is not applicable because there are no default authorization values.

High-Level Test Description
Findings: N/A

91 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.

High-Level Test Description
The evaluator logged into the TOE and disabled the key recovery option. The evaluator then confirmed that the user was not able to exercise key recovery. The evaluator then reinstalled the TOE with the 'noexport' option and confirmed that once the TOE was installed the user did not have access to the key recovery option.
Findings: PASS

92 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.

High-Level Test Description
This is not applicable because the TOE does not provide this functionality.
Findings: N/A

93 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.

High-Level Test Description
This is not applicable because the TOE does not provide this functionality.
Findings: N/A

3.2.3 AA: FMT_SMR.1 Security Roles

3.2.3.1 TSS

94 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.

3.2.3.2 Operational Guidance

95 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.

3.2.3.3 KMD

96 There are no KMD evaluation activities for this SFR.

3.2.3.4 Test

97 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.

3.3 Protection of the TSF (FPT)

3.3.1 AA:FPT_KYP_EXT.1 Protection of Key and Key Material

Technical Decision: The evaluation activities were modified per TD0458.

3.3.1.1 TSS

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

Findings: Section 6.5.1 of the [ST] states that the TOE only stores keys in non-volatile memory when encrypted as specified in FCS_COP.1(g). Section 6.2.11 states that key encryption is performed using AES-CBC-128 or AES-CBC-256.

3.3.1.2 Operational Guidance

99 There are no AGD evaluation activities for this SFR.

3.3.1.3 KMD

100 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.

Findings: [KMD] Section 3.1 provides Table 3: Key Lifecycle which lists all keys stored in non-volatile memory and how they are wrapped or encrypted.

3.3.1.4 Test

101 There are no test evaluation activities for this SFR.

3.3.2 AA:FPT_PWR_EXT.1 Power Saving States

3.3.2.1 TSS

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

Findings: Section 6.5.2 of the [ST] lists S4, G2(S5) and G3 as the compliant power saving states. The S4 power saving state is only supported on Windows platforms.

3.3.2.2 Operational Guidance

103 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.

Findings:	Section 2.5 of the [AGD] provides a list of compliant power saving states. The TOE itself does not offer any ability to configure power-saving states. Users interact with the Host OS or hardware platform to enter the above power states. The S4 power saving state is only supported on Windows platforms. Section 2.1 of the [AGD] offers additional instruction in which the 'Sleep' power saving state should be disabled in the host OS.

3.3.2.3 KMD

104 There are no KMD evaluation activities for this SFR.

3.3.2.4 Test

105 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).

High-Level Test Description
This was conducted as part of FCS_CKM.4(d).
Findings: PASS

3.3.3 AA:FPT_PWR_EXT.2 Timing of Power Saving States

3.3.3.1 TSS

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

Findings:	Section 6.5.3 of the [ST] states that the TOE enters a compliant power saving state when prompted by the protected OS and by user-initiated requests.

3.3.3.2 Operational Guidance

107 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.

Findings:	Section 2.5 of the [AGD] lists the compliant power saving states. The TOE itself does not offer any ability to configure power-saving states. Users are able to interact with the Host OS or hardware platform to enter the above power states. Power-loss is also explicitly denoted in section 2.5 of the [AGD] as being equivalent to power state G3, which is a supported compliant power-saving state.

3.3.3.3 KMD

108 There are no KMD evaluation activities for this SFR.

3.3.3.4 Test

109 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).

High-Level Test Description
This was conducted as part of FCS_CKM.4(d).
Findings: PASS

3.3.4 AA:FPT_TUD_EXT.1 Trusted Update

3.3.4.1 TSS

110 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.

Findings:	Section 6.5.4 of the [ST] states the update files obtained from the KLC web portal are digitally signed via RSA per FCS_COP.1(a) by KLC Group. The update is verified by the TOE prior to installation. The evaluator has determined that because the digital signature comes from the KLC Group, which is the TOE vendor, that it is considered an authorized source. All update files automatically have their digital signatures verified by the TOE prior to installation using the digital signature provided in the SecurityToken file that is distributed with the downloaded update package. Update files for EE are also obtained securely from the KLC customer web portal and contain a digital signature embedded within the binary.

111 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.

Findings:	Not applicable because the OE does not perform signature verification.

3.3.4.2 Operational Guidance

112 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.

Findings:	Section 2.7 of the [AGD] provides information on how the TOE acquires updates and that these updates are verified via digital signature. If signature verification fails, the update is aborted and an error message is displayed: “PBA Upgrade has failed”. Optionally, this section also includes a pointer to [MAN] section ‘*CDO Kryptr Upgrade via CLI*’ to performed the upgrade using the CLI rather than the UI.

3.3.4.3 KMD

113 There are no KMD evaluation activities for this SFR.

3.3.4.4 Test

114 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):

115 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.

High-Level Test Description
This test is performed in EE:FPT_TUD_EXT.1 Test 2.
Findings: PASS

116 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.

High-Level Test Description
This test is performed in conjunction with EE:FPT_TUD_EXT.1 Test 2.
Findings: PASS

4 Evaluation Activities for Optional Requirements (CPP_FDE_AA_V2.0E)

No evaluation activities for Optional Requirements from the CPP_FDE_AA_V2.0E have been selected.

5 Evaluation Activities for Selection-Based Requirements (CPP_FDE_AA_V2.0E)

5.1 Cryptographic Support (FCS)

5.1.1 AA: FCS_CKM.1(b) Cryptographic Key Generation (Symmetric Keys)

117 TSS

118 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.

Findings:	[ST] Section 6.2.3 lists the 256-bit AES keys. The TOE also generates 128-bit or 256-bit MK/BEVs based on the license applied during install. The figures in section 6.1.2 demonstrate how each of these keys are protected.

5.1.1.1 Operational Guidance

119 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.

Findings:	Section 2.8 of the [AGD] states that the TOE supports both 128 bit and 256 bits keys. The product license determines the support length of key. [MAN] sections '*CDO Kryptr License'*, '*Generate License Request and Import/Upgrade License',* and *'Generate a License Request File'* describe how to request a license and apply a license to the TOE.

5.1.1.2 KMD

120 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.

Findings:	Section 3.4 of the [KMD] demonstrates each key chain and how symmetric keys are used throughout the chain.

121 Test

122 There are no test evaluation activities for this SFR.

5.1.2 AA/EE: FCS_COP.1(a) Cryptographic Operation (Signature Verification)

123 This requirement is used to verify digital signatures attached to updates from the TOE manufacturer before installing those updates on the TOE. Because this component is to be used in the update function, additional Evaluation Activities to those listed below are covered in other evaluation activities sections in this document. The following activities deal only with the implementation for the digital signature algorithm; the evaluator performs the testing appropriate for the algorithm(s) selected in the component.

124 Hash functions and/or random number generation required by these algorithms must be specified in the ST; therefore the Evaluation Activities associated with those functions are contained in the associated Cryptographic Hashing and Random Bit Generation sections. Additionally, the only function required by the TOE is the verification of digital signatures. If the TOE generates digital signatures to support the implementation of any functionality required by this cPP, then the applicable valuation and validation scheme must be consulted to determine the required evaluation activities.

5.1.2.1 TSS

125 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).

Findings:	Section 6.2.8 of the [ST] states that the TOE performs signature verification using RSA 3072 with SHA-384 for trusted updates as follows: a) TOE updates are signed with the KLC code signing private key b) The obfuscated public key is embedded in the TOE binary c) When the user triggers the TOE update from the GUI, the TOE verifies the digital signature using the embedded public key d) If the digital signature verification succeeds, the upgrade process is carried out e) If the digital signature verification fails, the upgrade process is aborted, and an error is displayed to the user.

5.1.2.2 Operational Guidance

126 There are no AGD evaluation activities for this SFR.

5.1.2.3 KMD

127 There are no KMD evaluation activities for this SFR.

5.1.2.4 Test

128 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.

129 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.

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

131 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.

132 **ECDSA Algorithm Tests**

133 **ECDSA FIPS 186-4 Signature Verification Test**

134 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.

135 **RSA Signature Algorithm Tests**

136 **Signature Verification Test**

137 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.

138 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.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

5.1.3 AA: FCS_COP.1(b) Cryptographic Operation (Hash Algorithm)

5.1.3.1 TSS

139 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.

Findings:	Section 6.2.9 of the [ST] states that the TOE makes use of SHA-384 for digital signature verification and PBKDF.

5.1.3.2 Operational Guidance

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

Findings:	Only one hash size has been selected for AA:FCS_COP.1(b). No other hash sizes are configurable.

141

5.1.3.3 KMD

142 There are no KMD evaluation activities for this SFR.

5.1.3.4 Test

143 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.

144 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.

145 Short Messages Test Bit-oriented Mode

146 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.

147 Short Messages Test Byte-oriented Mode

148 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.

149 Selected Long Messages Test Bit-oriented Mode

150 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 $1 \leq i \leq m$. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 99*i, where $1 \leq i \leq m$. 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.

151 Selected Long Messages Test Byte-oriented Mode

152 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 $1 \leq i \leq m/8$. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 8*99*i, where $1 \leq i \leq m/8$. 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.

153 Pseudorandomly Generated Messages Test

154 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-ValidationProgram/documents/shs/SHAVS.pdf). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

5.1.4 AA: FCS_COP.1(c) Cryptographic Operation (Keyed Hash Algorithm)

5.1.4.1 TSS

155 If HMAC was selected:

156 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.

Findings:	Section 6.2.10 of the [ST] states that the TOE implements HMAC-SHA-384 with the characteristics of a key length of 384 bits, a block size of 1024 bits and a MAC length of 384 bits.

157 If CMAC was selected:

158 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.

Findings:	CMAC is not selected and is not applicable.

5.1.4.2 Operational Guidance

159 There are no AGD evaluation activities for this SFR.

5.1.4.3 KMD

160 There are no KMD evaluation activities for this SFR.

5.1.4.4 Test

161 If HMAC was selected:

162 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.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

163 If CMAC was selected:

164 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.

Findings:	This is not applicable since CMAC is not selected.

5.1.5 AA: FCS_COP.1(g) Cryptographic Operation (Key Encryption)

5.1.5.1 TSS

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

Findings:	Section 6.2.11 of the [ST] states that the TOE performs key encryption using AES-CBC-128 or AES-CBC-256.

5.1.5.2 Operational Guidance

166 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.

Findings:	Only a single mode of key encryption is supported; therefore, there is no configuration for switching between modes. [AGD] Section 2.8 states that both 256-bit and 128-bit DEKs and BEVs are supported. [MAN] Section 'CDO Kryptr License' describes how to request and import/upgrade license. Subsection 'Upgrade License' also states that the license file determines the key size used with the default being 256 bits unless a special request is made.

5.1.5.3 KMD

167 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.

Findings:	[KMD] Section 3.4 demonstrates each key chain and details how key encryption is used throughout the key chains.

5.1.5.4 Test

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

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

5.1.6 AA: FCS_KDF_EXT.1 Cryptographic Key Derivation

5.1.6.1 TSS

169 The evaluator shall verify the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 and SP 800-132.

Findings:	Section 6.2.12 of the [ST] states that passwords are conditioned via PBKDF2 using HMAC-SHA-384 with 100,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132.

5.1.6.2 Operational Guidance

170 There are no AGD evaluation activities for this SFR.

5.1.6.3 KMD

171 The evaluator shall examine the vendor's KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived.

Findings:	Section 3.4 of the [KMD] shows the keychains for password and dual-factor authentication which label the derivation (PBKDF2) which aligns with AA: FCS_KDF_EXT.1. Additionally, section 3.4.4 describes how PBKDF2 is used.

5.1.6.4 Test

172 There are no test evaluation activities for this SFR.

5.1.7 AA: FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

5.1.7.1 TSS

173 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.

Findings:	Section 6.2.14 of the [ST] states that the TOE implements a configurable password policy with the following options: a) Minimum Length (8 – 128) b) Require at least one uppercase c) Require at least one lowercase d) Require at least one numeric e) Require at least one special character (“!”,“”“,“#”,“$”,“%”,“&”,“‘”,“(”,“)”,“*”,“+”,“,”,“-”,“.”,“/”,“:”,“;”,“<”,“=”,“>”,“?”,“@”,“[”,“]”,“^”,“_”,“`”,“{”,“|”,“}”,“~”,“-”) f) History (Can repeat same password after configurable number of times) g) Number of consecutive failed validation attempts before reboot is required Passwords are conditioned via PBKDF2 using HMAC-SHA-384 with 100,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132.

5.1.7.2 Operational Guidance

174 There are no AGD evaluation activities for this SFR.

5.1.7.3 KMD

175 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.

Findings:	Figures within section 3.4 of the [KMD] display the TOE uses PBKDF resulting in a 256-bit key for password conditioning. This correlates with the key size selected in the [ST].

176 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.

Findings:	[KMD] Section 3.4.4 describes the use of PBKDF2 in detail, including the options, output key length and standard in which is conforms to. This section also describes how the password is concatenated with a salt prior to being passed to the PBKDF2 function. The figures in section 3.4 show how the resulting 256-bit submask is used to derive the BEV.

5.1.7.4 Test

177 The evaluator shall also perform the following tests:

178 Test 1: Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.

High-Level Test Description
The evaluator changed the password to a new one that is 64 characters long. Then, the evaluator tried to log in using the previous password resulting in an unsuccessful login. The evaluator logged in again with the new 64-character long password and verified that the TOE logged in successfully.
Findings: PASS

179 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.

High-Level Test Description
The evaluator logged into the TOE and attempted to change the password for a user to be 129 characters long and confirmed that the change attempt failed. The evaluator then attempted to log into the TOE with the 129 character password and confirmed access was denied. The evaluator than confirmed that the password set before attempting the 129 character password change was accepted.
Findings: PASS

180 Test 3: Ensure that the TOE supports passwords consisting of all characters assigned and supported by the ST author.

High-Level Test Description
The evaluator changed the password to a new one where all the supported characters are present. Then, the evaluator logged in using that password. The log in was successful.
Findings: PASS

5.1.8 AA: FCS_RBG_EXT.1 Random Bit Generation

5.1.8.1 TSS

181 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.

Findings:	Section 6.2.15 of the [ST] states that the BoringSSL cryptographic library is used in supported of AA cryptographic operations. BoringSSL implements CTR-DRBG(AES 256) seeded with 256-bits of entropy. The source of entropy for the TOE is the Intel DRNG (RDRAND). This is consistent with AA:FCS_RBG_EXT.1.2.

5.1.8.2 Operational Guidance

182 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.

Findings:	The TOE uses the DRBG listed in AA:FCS_RBG_EXT.1 with no further configuration necessary. Therefore, this is N/A.

5.1.8.3 KMD

183 There are no KMD evaluation activities for this SFR.

5.1.8.4 Test

184 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.

185 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).

186 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 foreach 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.

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

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

189 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.

190 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.

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

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

5.1.9 AA: FCS_SMC_EXT.1 Submask Combining

5.1.9.1 TSS

192 If the submasks produced from the authorization factors are XORed together to form the BEV or intermediate key, the TSS section shall identify how this is performed (e.g., if there are ordering requirements, checks performed, etc.). The evaluator shall also confirm that the TSS describes how the length of the output produced is at least the same as that of the BEV.

Findings:	Section 6.2.16 of the [ST] states that ASSKpass and ASSKsc are XORed together to form the intermediate key ASK that is then used with dual factor authentication configurations. No other ordering requirements are applicable. The output key length is at least the length of the BEV.

5.1.9.2 Operational Guidance

193 There are no AGD evaluation activities for this SFR.

5.1.9.3 KMD

194 The evaluator shall review the KMD to ensure that an approved combination is used and does not result in the weakening or exposure of key material.

Findings:	Section 3.4.2 item b) and subitem ii) of the [KMD] shows how the keys are combined for dual-factor authentication. This operation does not weaken or expose the key material.

5.1.9.4 Test

195 The evaluator shall perform the following test:

196 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 encrypted data.

High-Level Test Description
This test was performed as part of AA:FCS_AFA_EXT.1 and AA:FCS_AFA_EXT.2.
Findings: PASS

6 Evaluation Activities for SFRs (CPP_FDE_EE_V2.0E)

6.1 Cryptographic Support (FCS)

6.1.1 EE: FCS_CKM.1(c) Cryptographic Key Generation (Data Encryption Key)

6.1.1.1 TSS

1 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).

Findings:	Section 6.2.18 of the [ST] states that the TOE invokes the DRBG in AA:FCS_RBG_EXT.1 to generate a DEK during installation.

2 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.

Findings:	Section 6.2.18 of the [ST] states that the TOE generates the DEK directly from the DRBG.

3 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.

Findings:	The TOE does not receive the DEK from outside the host platform.

6.1.1.2 Operational Guidance

4 There are no AGD evaluation activities for this SFR.

6.1.1.3 KMD

5 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.

Findings:	The TOE does not receive the DEK from outside the host platform.

6.1.1.4 Test

6 The evaluator shall perform the following tests:

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

High-Level Test Description
No additional testing is necessary. [ST] claims both 128-bit and 256-bit configurations however, the configuration is dependent on the license file provided at install. The tests for FCS_CKM.4(d) were performed for both 128-bit and 256-bit installs and the evidence shows the appropriate keychain was configured.
Findings: PASS

6.1.2 EE: FCS_CKM.4(a) Cryptographic Key Destruction (Power Management)

6.1.2.1 TSS

8 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.

Findings:	[ST] Section 6.2.19 states that transitioning to a compliant power saving state will trigger the destruction of keys or key material in volatile memory. Users initiate a request to enter a power saving state by interaction with the Host OS. The TOE instructs the Protected OS to destroy keys and key material from volatile memory without delay when transitioning to a compliant power-saving state. Temporary keys and the allocated memory in which they reside are tracked during the usage lifecycle until destruction.

6.1.2.2 Operational Guidance

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

Findings:	Section 2.4 of the [AGD] states that the TOE handles the destruction of cryptographic keys and key material. When a user initiates a request to enter a power saving state, the TOE will instruct the protected OS to destroy all cryptographic keys and key material from volatile memory.

6.1.2.3 KMD

10 The evaluator shall check to ensure the KMD lists each type of key, its origin, possible memory locations in volatile memory.

Findings:	Table 3: ‘Key Lifecycle’ in the [KMD] lists the derivation and the storage location of all keys.

6.1.2.4 Test

11 There are no test evaluation activities for this SFR.

6.1.3 EE: FCS_CKM_EXT.4(a) Cryptographic Key and Key Material Destruction (Destruction Timing)

6.1.3.1 TSS

12 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.

Findings:	[ST] Section 6.2.20 states that the TOE destroys all keys and keying material in volatile memory when no longer needed, including when the TOE transitions into a compliant power-saving state or is shut down, and when keys are overwritten with new ones. [KMD] Section 3.1 also contains Table 3: Key Lifecycle which details the exact conditions in which each key is destroyed.

6.1.3.2 Operational Guidance

13 There are no AGD evaluation activities for this SFR.

6.1.3.3 KMD

14 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.

Findings:	[KMD] Table 3: 'Key Lifecycle' lists all keys. The column labelled 'End-of-Life/When key is destroyed' describes when the key is no longer needed.

15 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

Findings:	[KMD] Table 3: 'Key Lifecycle' lists all keys. The column labelled 'End-of-Life/When key is destroyed' describes when the key is no longer needed. Section 3.5 provides key destruction methods for key material. Keys are erased which follows the selection for EE:FCS_CKM.4(a).

6.1.3.4 Test

16 There are no test evaluation activities for this SFR.

6.1.4 EE: FCS_CKM_EXT.4(b) Cryptographic Key and Key Material Destruction (Power Management)

6.1.4.1 TSS

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

Findings:	Section 6.2.21 of the [ST] states that when the TOE transitions to a compliant power saving state, all key material, BEV, and authentication factors stored in plaintext are destroyed.

6.1.4.2 Operational Guidance

18 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.

Findings:	Section 2.1 of the [AGD] instructs that the 'Sleep' power saving state should be disabled in the host OS. 'Sleep' is a non-compliant power saving state and is not included in EE:FPT_PWR_EXT.1. Section 2.5 also states that an unexpected power loss would result in the G3 power state. Therefore, there are no instances where the TOE may enter a non-compliant power saving state.

6.1.4.3 KMD

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

Findings:	Table 3: 'Key Lifecycle' in [KMD] lists the storage location of all keys.

20 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.

Findings:	Table 3: 'Key Lifecycle' in [KMD] lists where key material resides, its purpose and how it is destroyed. EE:FCS_CKM_EXT.6 claims that the TSF shall use methods stated in EE:FCS_CKM.4(d) for key destruction. The evaluator has confirmed that keys and key material are erased.

6.1.4.4 Test

21 There are no test evaluation activities for this SFR.

6.1.5 EE: FCS_CKM_EXT.6 Cryptographic Key Destruction Types

6.1.5.1 TSS/KMD (Key Management Description may be used if necessary details describe proprietary information)

22 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.

Findings:	[ST] Section 6.2.23 states that the TOE implements the key destruction methods and types described in FCS_CKM.4(d). The evaluator reviewed [KMD] Section 3.1 Table 3: Key Lifecycle and confirmed all keys subject to destruction by the EE component use a method specified in FCS_CKM.4(d).

6.1.5.2 Operational Guidance

23 There are no AGD evaluation activities for this SFR.

6.1.5.3 Test

24 There are no test evaluation activities for this SFR.

6.1.6 EE: FCS_KYC_EXT.2 Key Chaining (Recipient)

6.1.6.1 TSS

25 There are no TSS evaluation activities for this SFR.

6.1.6.2 Operational Guidance

26 There are no AGD evaluation activities for this SFR.

6.1.6.3 KMD

27 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).

Findings:	Section 3.4 of the [KMD] provides information on the hierarchy and details of the keychain. The evaluator confirmed that the key encryption (AES-CBC-128 or AES-CBC-256) and key derivation (PBKDF2) are consistent with EE:FCS_COP.1(g) and EE:FCS_KDF_EXT.1, respectively. [ST] does not make claims for FCS_COP.1(d) or FCS_COP.1(e) for the EE component of the TOE.

28 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.

Findings:	Section 3.4 of the [KMD] is dedicated to keychains. This section provides the key management aspect of the TOE as well as figures of the keychains. These figures include the key hierarchy. Table 3 of the [KMD] is the most comprehensive display of storage location for all keys. Key derivation is listed under the “Derivation” column in Table 3. After examining the key hierarchy the evaluator found no way the DEK could be decrypted without cryptographic exhaust or knowledge of the BEV. The effective strength of the DEK is maintained throughout the key chain.

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

Findings:	[KMD] Table 3 provides details for all keys. Each key is annotated with the strength of the key.

6.1.6.4 Test

30 There are no test evaluation activities for this SFR.

6.1.7 EE: FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

6.1.7.1 TSS

31 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.

Findings:	Section 6.2.31 of the [ST] states that salts are generated by the DRBG provided by the host platform using the *gcry_randomize* function.

32 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.

Findings:	Section 6.2.31 of the [ST] states that the tweak values start from a non-negative integer that is incremented by sector number for both Windows and Linux hosts. IVs are not used with AES-XTS. The TOE does not make use of nonces.

6.1.7.2 Operational Guidance

33 There are no AGD evaluation activities for this SFR.

6.1.7.3 KMD

34 There are no KMD evaluation activities for this SFR.

6.1.7.4 Test

35 There are no test evaluation activities for this SFR.

6.1.8 EE: FCS_VAL_EXT.1 Validation

6.1.8.1 TSS

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

Findings:	Section 6.2.32 states that the TOE validates the BEV. Per section 6.1.2 both authorization factors result in an FSKEY to create the BEV.

37 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).

Findings:	[ST] Section 6.2.32 states that the TOE validates the BEV by using it to derive an intermediate key using PBKDF2. This intermediate key is used to decrypt a known value which is compared to a stored value. Per the keychain in section 6.1.2 this can only be performed after the submasks are combined.

38 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.

Findings:	Section 6.2.32 of the [ST] states that successful validation of BEV is required prior the decryption of the drive and granting access to any TSF data after exiting a compliant power saving state. The evaluator determined that both authorization factors can be used to exit a compliant power saving state per section 6.2.2.

6.1.8.2 Operational Guidance

39 (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.

Findings:	[AGD] Section 2.11 states that the failure threshold can be configured between 1 and 20 attempts by the “Administrator’ or ‘Security Officer’ via the ‘Settings > Configuration > Security’ menu.

40 (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.

Findings:	[AGD] Section 2.11 states that the validation attempts can be configured between 1 and 20.

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

Findings:	[AGD] Section 2.5 states “When resuming from the above power saving states, users are required to re-authenticate using the same authorization factors as per normal operation.”

6.1.8.3 KMD

42 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.

Findings:	This activity is completed via information from [ST] and [KMD]. [ST] Section 6.2.32 states that the TOE validates the BEV prior to allowing decryption and granting access to any TSF data after exiting a compliant power saving state. If the validation is unsuccessful, the TOE increments a failed authentication counter. After the configured number of failed BEV validation/authentication attempts is reached, the user will be locked out until the system is rebooted. The evaluator confirmed this is consistent with the description in [KMD] section 2.1.1.

43 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.

Findings:	Section 6.2.32 of the [ST] states that the TOE validates the BEV by using it to derive an intermediate key using PBKDF2 which is then used to decrypt a known value which is compared to a stored known value. This is consistent with the EE:FCS_VAL_EXT.1 selection to validate the “intermediate key”.

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

Findings:	See previous finding. The evaluator had checked the figures in section 3.4 of the [KMD] and found that the process does not compromise any material that might expose the submask.

6.1.8.4 Test

45 The evaluator shall perform the following tests:

46 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.

High-Level Test Description
The evaluator configured the TOE to accept 4 incorrect login attempts before locking the user out. The evaluator then failed to log in four times and ensured that when the fourth incorrect attempt is reached, the TOE did not allow the user to log in a fifth time even when providing the correct password. The evaluator then power cycled the TOE and confirmed that the correct password was accepted after reboot. The test was repeated using double factor authentication as the login method.
Findings: PASS

47 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.

High-Level Test Description
This test has been satisfied in Test 1 above and FCS_AFA_EXT.2 testing.
Findings: PASS

6.2 User Data Protection (FDP)

6.2.1 EE: FDP_DSK_EXT.1 Protection of Data on Disk

6.2.1.1 TSS

48 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.

Findings:	[ST] Sections 6.3.1.1 and 6.3.1.2 state that the Encryption Engine installer or the CDO installer use cryptsetup and CDOcryptsetup respectively for initial encryption. During runtime, the dm-crypt disk encryption subsystem (Linux) and CDO drivers (Windows) perform the encryption/decryption. Section 6.3.1 of the [ST] states that the TOE encrypts all protected data using AES without user intervention. All standard methods of accessing a protected disk drive via the host platform’s operating system will pass through these functions without exception.

49 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.

Findings:	Section 6.3.1.1 of the [ST] states that the TOE uses dm-crypt in the Linux kernel for Linux encryption/decryption during runtime. All other encryption/decryption is handled by the TOE.

50 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.

Findings:	[ST] Sections 6.3.1.1 and 6.3.1.2 state that the Encryption Engine installer or the CDO installer use cryptsetup and CDOcryptsetup respectively for initial encryption. During runtime, the dm-crypt disk encryption subsystem (Linux) and CDO drivers (Windows) perform the encryption/decryption. Section 6.3.1 states that the TOE encrypts all protected data using AES without user intervention. All protected data is written to disk in blocks per standard operating conditions of the AES algorithm.

51 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.

Findings:	[ST] Section 6.3.1 states that the TOE performs full drive encryption in accordance with FCS_COP.1(f) on all disk drives specified at installation, such that the entire drive(s) are encrypted and contain no plaintext protected data. All partitions are encrypted except the ESP partition on Red Hat systems (Note: ESP partition is not used post installation). Section 6.3.1 states that the TOE encrypts all protected data using AES without user intervention.

6.2.1.2 Operational Guidance

52 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.

Findings:	[AGD] Section 2.1 describes the steps to prepare the device for installation of the TOE. Section 2.2 also references [MAN] section '*Installation of CDO Kryptr'* for installation of the TOE. The evaluator followed the steps when installing the TOE and found them to be sufficient to ensure that all hard drive devices will be encrypted.

6.2.1.3 KMD

53 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.

Findings:	[KMD] Section 2 describes the encryption engine as a module that provides full disk encryption for Linux and Windows operating systems. Section 2.2 includes Encryption Engine/Driver information for both Linux and Windows along with a description of the encryption/decryption process for each operating system in subsections 2.2.1 and 2.2.2. No hardware encryption diagrams are applicable since the TOE provides software encryption.

54 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).

Findings:	[KMD] Section 2.2.1.1 describes the encryption process for Linux and the partitions exempt from encryption. Section 2.2.2.1 also describes the encryption process for Windows.

55 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.

Findings:	[KMD] Section 2.1 describes the TOE boot/pre-boot authentication process as well as how the AA interacts with the EE. Section 2.2 also describes the EE driver for both Windows and Linux which includes the point in which encryption and decryption occur.

6.2.1.4 Test

56 The evaluator shall perform the following tests:

57 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.

High-Level Test Description
A random 64KB character pattern was created and stored in the TOE's Host OS and the partition's highest and lowest logical addresses were determined.
Findings: PASS

58 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.

High-Level Test Description
Two more 64KB files were created and stored on the TOE's Host OS.
Findings: PASS

59 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.

High-Level Test Description
The evaluator analyzed the drive in an unencrypted state and located the pattern for each file on disk. The evaluator then analyzed the encrypted drive at the logical locations where the file was stored and confirmed that pattern for each file was not found. Note: These steps were performed in reverse of the steps in the AA outlined above so that the evaluator could establish the location of the data when the drive was unencrypted.
Findings: PASS

6.3 Security management (FMT)

6.3.1 EE:FMT_SMF.1 Specification of Management Functions

6.3.1.1 TSS

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

Findings:	Section 6.4.4 of the [ST] states that the TOE performs changes to the DEK for each environment: a) **Linux.** On user's request, CDO sends luksAddKey and luksKillSlot commands to cryptsetup engine using the Admin AK to destroy the keys. b) **Windows.** On user's request, CDO sends ChangeDEK command to CDOCryptsetup engine using the Admin AK. The engine then wipes the key area with random data before deleting it.

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

Findings:	Section 6.4.4 of the [ST] states that the TOE erases the DEK per section 6.4.2: a) **Linux.** On user's request for cryptographic erase, luksAddKey command is sent to the cryptsetup engine using the Admin AK with the value of a new random key followed by luksKillSlot for Admin and User slots. b) **Windows**. On user's request for cryptographic erase, CDO sends ChangeDEK command to CDOCryptsetup engine using the Admin AK

62 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.

Findings:	Section 6.4.4 of the [ST] states that TOE updates are initiated manually via the CDO Kryptr UI.

63 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.

Findings:	Item d) does not claim any additional functions.

6.3.1.2 Operational Guidance

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.

Findings:	Section 2.6 of the [AGD] references the section labelled "Change DEK" in the [MAN] which provides instructions for changing the DEK. This applies to all environments on which the TOE is claiming conformance.

64 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.

Findings:	Section 2.7 of the [AGD] provides information on how the TOE can be updated. Software updates files must have to be manually downloaded from the KLC web portal and then copied to a USB drive. Detailed Update instructions are provided in [MAN] section "CDO Kryptr Upgrade".

65 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.

Findings:	This is N/A since item d) is not selected in EE:FMT_SMF.1.1.

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

Findings:	This is N/A since this is not selected in EE:FMT_SMF.1.1.

6.3.1.3 KMD

67 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.

Findings:	Not applicable because item d) is not selected in EE:FMT_SMF.1.1.

6.3.1.4 Test

68 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).

High-Level Test Description
Item a) and b) are covered in AA:FMT_SMF.1 a) and b).
Findings: PASS

69 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.

High-Level Test Description
Item c) is covered in EE:FPT_TUD_EXT.1.
Findings: PASS

70 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.

High-Level Test Description
Item d) is N/A, no additional functions are claimed.
Findings: N/A

6.4 Protection of the TSF (FPT)

6.4.1 EE: FPT_KYP_EXT.1 Protection of Key and Key Material

Technical Decision: The evaluation activities were modified per TD0458.

6.4.1.1 TSS

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

Findings:	Section 6.5.1 of the [ST] states that the TOE only stores keys in non-volatile memory when encrypted as specified in FCS_COP.1(g).

6.4.1.2 Operational Guidance

72 There are no AGD evaluation activities for this SFR.

6.4.1.3 KMD

73 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.

Findings:	Table 3 in the [KMD] lists the storage location and protection of all keys. Where applicable, keys are protected using AES-CBC-128 or AES-CBC-256 which is consistent with the EE:FPT_KYP_EXT.1 selection (EE:FCS_COP.1(g)).

6.4.1.4 Test

74 There are no test evaluation activities for this SFR.

6.4.2 EE: FPT_PWR_EXT.1 Power Saving States

6.4.2.1 TSS

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

Findings:	Section 6.5.2 of the [ST] lists S4, G2(S5) and G3 as the compliant power saving states. The S4 power saving state is only supported on Windows platforms.

6.4.2.2 Operational Guidance

Technical Decision: The evaluation activities were modified per TD0460.

76 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.

Findings:	The [AGD] lists the power saving states in section 2.5. These include S4, G2 (S5) and G3. The S4 power saving state is only supported on Windows platforms. Section 2.1 of the [AGD] offers additional instruction in which the ‘Sleep’ power saving state should be disabled in the host OS.

6.4.2.3 KMD

77 There are no KMD evaluation activities for this SFR.

6.4.2.4 Test

78 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.

High-Level Test Description
This was conducted as part of AA:FCS_CKM.4(d).
Findings: PASS

6.4.3 EE: FPT_PWR_EXT.2 Timing of Power Saving States

6.4.3.1 TSS

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

Findings:	Section 6.5.3 of the [ST] states that the TOE enters a compliant power saving state when prompted by the protected OS and by user-initiated requests.

6.4.3.2 Operational Guidance

80 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).

Findings:	[AGD] Section 2.5 states that the TOE will enter a compliant power saving state immediately when prompted by the protected OS after a user-initiated request. This is expected to complete within several seconds, dependent on the host OS.

6.4.3.3 KMD

81 There are no KMD evaluation activities for this SFR.

6.4.3.4 Test

82 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.

High-Level Test Description
This was conducted as part of AA:FCS_CKM.4(d).
Findings: PASS

6.4.4 EE: FPT_TST_EXT.1 TSF Testing

6.4.4.1 TSS

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

Findings:	Section 6.5.5 of the [ST] lists the libgcrypt Known Answer Tests (KAT) that are performed at power-on. The evaluator confirmed that the tests cover the claimed cryptographic functions.

84 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.

Findings:	In addition to the KAT crypto self tests, Section 6.5.5 of the [ST] includes integrity tests for the AA and EE. If the tests fail, the protected OS will not boot.

85 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.

Findings:	[ST] Section 6.2.30 describes the health tests that are implemented and are consistent with NIST SP 800-90A Section 11.3.

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

Findings:	Section 6.5.5 of the [ST] lists the libgcrypt Known Answer Tests (KAT) that are performed at power-on. The evaluator confirmed that the tests cover the claimed FCS_COP functions: EE:FCS_COP.1(b): GCRY_MD_SHA256 for SHA-256 and GCRY_MD_SHA384 for SHA-384 EE:FCS_COP.1(c): GCRY_MAC_HMAC_SHA256 for HMAC-SHA-256 and GCRY_MAC_HMAC_SHA384 for HMAC-SHA-384 EE:FCS_COP.1(f): GCRY_CIPHER_AES128 and GCRY_CIPHER_AES256 for AES-XTS 128 and AES-XTS 258, respectively. EE:FCS_COP.1(g): GCRY_CIPHER_AES128 for AES-CBC 128 and GCRY_CIPHER_AES256 for AES-CBC 256

87 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 startup.

Findings:	In addition to the KAT crypto self tests, Section 6.5.5 of the [ST] includes integrity tests for the AA and EE. If the tests fail, the protected OS will not boot.

6.4.4.2 Operational Guidance

88 There are no AGD evaluation activities for this SFR.

6.4.4.3 KMD

89 There are no KMD evaluation activities for this SFR

6.4.4.4 Test

90 There are no test evaluation activities for this SFR.

6.4.5 EE: FPT_TUD_EXT.1 Trusted Update

6.4.5.1 TSS

91 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.

Findings:	Section 6.5.4 of the [ST] states the update files are digitally signed (RSA per FCS_COP.1(a)) by KLC Group and verified prior to installation. The evaluator has determined that because the digital signature comes from the KLC Group, which is the TOE vendor, that it is considered an authorized source. All update files automatically have their digital signatures verified by the TOE prior to installation using the digital signature provided in the SecurityToken file that is distributed with the downloaded update package. Update files for EE are also obtained securely from the KLC customer web portal and contain a digital signature embedded within the binary.

92 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.

Findings:	Not applicable because the OE does not perform signature verification.

6.4.5.2 Operational Guidance

93 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.

Findings:	Section 2.7 of the [AGD] states that the software updates for the TOE must be manually downloaded from the vendor's web portal and then copied to a USB drive. The TOE's UI is used to trigger the update from the USB. The TOE is digitally signed and the signature is verified prior to install. If the signature verification fails, the update is abandoned and an error message is displayed. Optionally, this section also includes a pointer to [MAN] section '*CDO Kryptr Upgrade via CLI*' to performed the upgrade using the CLI rather than the UI.

6.4.5.3 KMD

94 There are no KMD evaluation activities for this SFR.

6.4.5.4 Test

95 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):

96 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.

High-Level Test Description
This test is performed in EE:FPT_TUD_EXT.1 Test 2.
Findings: PASS

97 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.

High-Level Test Description
The evaluator confirmed the current build of the TOE. The evaluator then upgraded the TOE to a newer build. When the update was completed, the evaluator verified that the TOE reported the new version was installed. The evaluator then repeated the following tests: AA:FCS_AFA_EXT.1 and AA:FMT_SMF.1 c) and confirmed that the update had not affected the functionality of the TOE.
Findings: PASS

7 Evaluation Activities for Optional Requirements (CPP_FDE_EE_V2.0E)

There are no optional CPP_FDE_EE_V2.0E requirements chosen for this evaluation.

8 Evaluation Activities for Selection-Based Requirements (CPP_FDE_EE_V2.0E)

8.1 Cryptographic Support (FCS)

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

8.1.1.1 TSS + KMD (Key Management Description may be used if necessary details describe proprietary information)

98 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.

Findings:	[KMD] is used to satisfy this activity. Section 3.1 provides Table 3: Key Lifecycle which lists all keys stored in volatile memory, how they are derived and/or wrapped, and their destruction method. For each key in volatile memory, this table describes how they are overwritten. The evaluator confirmed that this description is consistent with the AA:FCS_CKM.4(d) claims. [ST] Section 6.2.22 also states that on Linux crypt_safe_free is used to zeroize memory. On Windows, memory is zeroized before destruction.

99 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).

Findings:	[KMD] is used to satisfy this activity. Section 3.1 provides Table 3: Key Lifecycle which lists all keys stored in non-volatile memory, how they are derived and/or wrapped, and their destruction method. For each key in non-volatile memory, this table describes how they are destroyed. The evaluator confirmed that this description is consistent with the AA:FCS_CKM.4(d) claims.

100 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.

Findings:	[KMD] is used to satisfy this activity. Section 3.1. provides Table 3: Key Lifecycle for all keys. For each media type, the storage location and destruction method is consistent with the EE:FCS_CKM.4(d) selections and the TSS.

101 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.

Findings:	No circumstances are identified in which the TOE does not strictly conform to the key destruction requirement. [ST] does not make use of the open assignment.

102

8.1.1.2 Operational Guidance

103 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.

104 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.

105 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.

106 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.

107 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.

Findings:	Section 2.4 of the [AGD] states that the TOE handles the destruction of cryptographic keys and key material when they are no longer required. There are no situations where key destruction would be delayed or prevented. The TOE also instructs the protected OS to destroy all cryptographic keys and key material in volatile memory when entering a power saving state.

8.1.1.3 Test

Technical Decision: The evaluation activities were modified per TD0766.

108 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.

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

110 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.

High-Level Test Description
This test is performed in AA:FCS_CKM.4(d).
Findings: PASS

111 *The following tests apply only 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.*

112 *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.*

113 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 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 Use Case 1 test 1 above), and if a fragment is found in the repeated test then the test fails.

High-Level Test Description
The above test is not applicable since the ST selects "*instructs the underlying platform…"*.
Findings: N/A

114 *The following tests apply only 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.*

115 *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.*

116 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 storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

117 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.

High-Level Test Description
The above test is not applicable since the ST selects "*instructs the underlying platform…"*.
Findings: N/A

8.1.2 EE: FCS_COP.1(b) Cryptographic Operation (Hash Algorithm)

8.1.2.1 TSS

118 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.

Findings:	Section 6.2.24 of the [ST] states that the TOE uses SHA-384 for digital signature verification and PBKDF2. The TOE uses SHA-256 in the HMAC_DRBG in support of EE cryptographic operations from the PBA.

8.1.2.2 Operational Guidance

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

Findings:	No configuration is needed. [ST] Section 6.2.24 states that these algorithms are not configurable within the TOE.

8.1.2.3 KMD

120 There are no KMD evaluation activities for this SFR.

8.1.2.4 Test

121 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.

122 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.

123 Short Messages Test Bit-oriented Mode

124 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.

125 Short Messages Test Byte-oriented Mode

126 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.

127 Selected Long Messages Test Bit-oriented Mode

128 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.

129 Selected Long Messages Test Byte-oriented Mode

130 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.

131 Pseudorandomly Generated Messages Test

132 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-ValidationProgram/documents/shs/SHAVS.pdf). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

8.1.3 EE: FCS_COP.1(c) Cryptographic Operation (Keyed Hash Algorithm)

8.1.3.1 TSS

133 If HMAC was selected:

134 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.

Findings:	Section 6.2.25 of the [ST] notes that the TOE implements HMAC-SHA-256 and HMAC-SHA-384. HMAC-SHA-256 uses a key length of 256 bits, a block size of 512 bits, and a MAC length of 256 bits. HMAC-SHA-384 uses a key length of 384 bits, a block size of 1024 bits and a MAC length of 384 bits.

135 If CMAC was selected:

136 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.

Findings:	CMAC was not selected and is not applicable.

8.1.3.2 Operational Guidance

137 There are no AGD evaluation activities for this SFR.

8.1.3.3 KMD

138 There are no KMD evaluation activities for this SFR.

8.1.3.4 Test

139 If HMAC was selected:

140 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.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

141 If CMAC was selected:

142 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.

Findings:	This is not applicable since CMAC is not selected.

8.1.4 EE: FCS_COP.1(f) Cryptographic Operation (AES Data Encryption/Decryption)

8.1.4.1 TSS

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

Findings:	Section 6.2.26 of the [ST] states that disk encryption is performed using AES-XTS 256 by default for both Linux and Windows. AEX-XTS 128 is also supported on both platforms by configuring a specific license during installation.

8.1.4.2 Operational Guidance

144 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.

Findings:	This is N/A since multiple encryption modes are not supported.

8.1.4.3 KMD

145 There are no KMD evaluation activities for this SFR.

8.1.4.4 Test

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

147 **AES-CBC Tests**

148 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.

149 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.

150 **AES-CBC Known Answer Tests**

151 KAT-1 (GFSBox):

152 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.

153 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.

154 KAT-2 (KeySBox):

155 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.

156 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.

157 KAT-3 (Variable Key):

158 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.

159 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.

160 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.

161 KAT-4 (Variable Text):

162 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.

163 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.

164 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.

165 **AES-CBC Multi-Block Message Test**

166 The evaluator shall test the encrypt functionality by encrypting nine i-block messages for each selected key size, for $2 \leq i \leq 10$. 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.

167 The evaluator shall test the decrypt functionality by decrypting nine i-block messages for each selected key size, for $2 \leq i \leq 10$. 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.

168 **AES-CBC Monte Carlo Tests**

169 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.

170 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.

171 # 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]

}
```

172 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.

173 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.

174 **AES-GCM Test**

175 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.

176 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.

177 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.

178 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.

179 **XTS-AES Test**

180 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.

181 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.

182 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.

183 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.

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

8.1.5 EE: FCS_COP.1(g) Cryptographic Operation (Key Encryption)

8.1.5.1 TSS

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

Findings:	Section 6.2.27 of the [ST] states that both the Windows and Linux environment use AES-CBC for MK encryption and generate key sizes of 128 or 256 bits.

8.1.5.2 Operational Guidance

185 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.

Findings:	Only a single mode of key encryption is supported; therefore, there is no configuration for switching between modes. [AGD] Section 2.8 states that both 256-bit and 128-bit DEKs and BEVs are supported. [MAN] Section ‘CDO Kryptr License’ describes how to request and import/upgrade license. Subsection ‘Upgrade License’ also states that the license file determines the key size used with the default being 256 bits unless a special request is made.

8.1.5.3 KMD

186 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.

Findings:	Section 3.4.2 of the [KMD] discusses the KEKs and how they interact with the keychain.

8.1.5.4 Test

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

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

8.1.6 EE: FCS_KDF_EXT.1 Cryptographic Key Derivation

8.1.6.1 TSS

188 The evaluator shall verify the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 and SP 800-132.

Findings:	Section 6.2.28 of the [ST] states that the TOE utilizes PBKDF2 as its key derivation function with the parameters of HMAC-SHA-384 with 100,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132. This process is used by both cryptsetup in Linux and CDOCryptsetup in Windows.

8.1.6.2 Operational Guidance

189 There are no AGD evaluation activities for this SFR.

8.1.6.3 KMD

190 The evaluator shall examine the vendor's KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived.

Findings:	Table 3 of the [KMD] lists all keys. The individual key derivation method is listed under the 'Derivation' column. Section 3.4.2 also describes the KEKs. The evaluator confirmed that intermediate keys use an approved function per EE:FCS_COP.1(c).

8.1.6.4 Test

191 There are no test evaluation activities for this SFR.

8.1.7 EE: FCS_RBG_EXT.1 Random Bit Generation

8.1.7.1 TSS

192 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.

Findings:	Section 6.2.30 of the [ST] states that the Libgcrypt cryptographic library is used in supported of EE cryptographic operations from the PBA. Libgcrypt implements HMAC-DRBG(SHA-256) seeded with 384-bits of entropy. The source of entropy for the TOE is the Intel DRNG (RDRAND). This is consistent with EE:FCS_RBG_EXT.1.2

8.1.7.2 Operational Guidance

193 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.

Findings:	The TOE uses the DRBG listed in EE:FCS_RBG_EXT.1 with no further configuration necessary. Therefore, this is N/A.

8.1.7.3 KMD

194 There are no KMD evaluation activities for this SFR.

8.1.7.4 Test

195 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.

196 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).

197 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 foreach 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.

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

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

200 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.

201 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.

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

Findings:	This is covered by CAVP certificates as shown in Table 12 of the Security Target.

9 Evaluation Activities for SARs

9.1 Security Target (ASE)

9.1.1 ASE_CCL.1 Exact Conformance Actions

9.1.1.1 ASE_CCL.1.8C

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

Findings:	[ST] Section 3 includes the security problem definition from CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E. The statements of security definition are identical in the PP and the ST.

9.1.1.2 ASE_CCL.1.9C

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

Findings:	[ST] Section 4 includes the security objectives from CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E. The statements of security definition are identical in the PP and the ST.

9.1.1.3 ASE_CCL.1.10C

205 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.

Findings:	[ST] Section 5 includes the security requirements from the CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E. All mandatory SFRs in the cPP and all of the selection-based SFRs that are entailed by selections made are present in Section 5 of the [ST]. No optional SFRs are claimed.

9.2 Development (ADV)

9.2.1 Basic Functional Specification (ADV_FSP.1)

206 The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2 (Evaluation Activities for SFRs), and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

207 The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

208 The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

209 The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional "functional specification" documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7 is treated as implicit and no separate mapping information is required for this element.

9.2.1.1 ADV_FSP.1-1

210 The evaluator shall examine the functional specification to determine that it states the purpose of each SFR-supporting and SFR-enforcing TSFI.

211 Evaluation Activity: 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.

Findings:	The evaluator examined the [AGD] (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator verified the [AGD] describes the purpose and method of use for each security relevant TSFI by verifying the AGD satisfies all of the Guidance Evaluation Activities.

9.2.1.2 ADV_FSP.1-2

212 The evaluator shall examine the functional specification to determine that the method of use for each SFR-supporting and SFR-enforcing TSFI is given.

213 Evaluation Activity: 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.

Findings:	The evaluator examined the [AGD] (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator verified the [AGD] describes the purpose and method of use for each security relevant TSFI by verifying the [AGD] satisfies all of the Guidance Evaluation Activities.

9.2.1.3 ADV_FSP.1-3

214 The evaluator shall examine the presentation of the TSFI to determine that it identifies all parameters associated with each SFR-enforcing and SFR supporting TSFI.

215 Evaluation Activity: 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.

Findings:	The evaluator examined the [AGD] (interface documentation) to verify that it describes the parameters, purpose and method of use for each TSFI that is identified as being

	security relevant. The evaluator verified the [AGD] describes the parameters, purpose and method of use for each security relevant TSFI by verifying the [AGD] satisfies all of the Guidance Evaluation Activities.

9.2.1.4 ADV_FSP.1-4

216 The evaluator shall examine the rationale provided by the developer for the implicit categorisation of interfaces as SFR non-interfering to determine that it is accurate.

217 Paragraph 561 from the CEM: “In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, this work unit should be considered satisfied.”

218 Since the rest of the ADV_FSP.1 work units will have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.

Findings:	This work unit is covered with the rest of the ADV_FSP.1 work units.

9.2.1.5 ADV_FSP.1-5

219 The evaluator shall check that the tracing links the SFRs to the corresponding TSFIs.

220 Evaluation Activity: The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

Findings:	The evaluation team examined the interface documentation and was able to map interfaces to SFRs, sufficient to enable each of the evaluation activities to be completed satisfactorily. The evaluation team’s results from performing the evaluation activities are documented in Sections 3 through 8 of this AAR.

9.2.1.6 ADV_FSP.1-6

221 The evaluator shall examine the functional specification to determine that it is a complete instantiation of the SFRs.

222 EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are covered. Therefore, the intent of this work unit is covered.

Findings:	This work unit is covered with the EAs associated with the SFRs throughout this document.

223 ADV_FSP.1-7

224 The evaluator shall examine the functional specification to determine that it is an accurate instantiation of the SFRs.

225 EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are addressed, and that the description of the interfaces is accurate with respect to the specification captured in the SFRs. Therefore, the intent of this work unit is covered.

Findings:	This work unit is covered with the EAs associated with the SFRs throughout this document.

9.2.1.7 Evaluation Activity

226 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.

227 In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g., audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent, is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

228 The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

Findings:	The assurance activities from Supporting Documents of the CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E have been performed. The evaluator concluded adequate information was provided and the analysis of the evaluator is documented in Sections 3 through 8 of this document.

9.2.1.8 Evaluation Activity

229 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.

Findings:	The assurance activities from Supporting Documents of the CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E have been performed. The evaluator concluded adequate information was provided and the analysis of the evaluator is documented in Sections 3 through 8 of this document.

9.2.1.9 Evaluation Activity

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

231 The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2 (Evaluation Activities for SFRs), including the EAs associated with testing of the interfaces.

232 It should be noted that there may be some SFRs that do not have an interface that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

233 However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a ‘fail’.

Findings:	The assurance activities from Supporting Documents of the CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E have been performed. The evaluator concluded adequate information was provided and the analysis of the evaluator is documented in Sections 3 through 8 of this document.

9.3 Guidance Documents (AGD)

234 It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the Evaluation Activities in this section are described under the traditionally separate AGD families, the mapping between real TOE documents and AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to administrators and users (as appropriate) as part of the TOE.

9.3.1 Operational User Guidance (AGD_OPE.1)

235 Specific requirements and checks on the user guidance documentation are identified (where relevant) in the individual Evaluation Activities for each SFR, and for some other SARs (e.g. ALC_CMC.1).

9.3.1.1 Evaluation Activity:

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

237 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.

238 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.

239 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.

240 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.

Findings:	The evaluator checked the requirements above are met by the guidance documentation. The 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. The CC guidance will also be published on www.niap-ccevs.org. The evaluator ensured that the Operational Guidance is provided for every Operational Environment (OE) that the product supports as claimed in the Security Target. Section 1.3.2 ‘Evaluated Software’ of the [AGD] specifies the TOE and Section 1.3.3 of the [AGD] specifies the supported OE (Non-TOE Components).

The [AGD] Section 2.8 ‘Cryptography’ provides instructions for cryptographic configuration.

The [AGD] Section 1.3.4 ‘Evaluated Functions’ lists which security functionality is in the scope of the evaluation.

The evaluator verified the operational guidance documentation makes it clear which security functionality is covered by the Evaluation Activities.

9.3.2 Preparative Procedures (AGD_PRE.1)

241 As for the operational guidance, specific requirements and checks on the preparative procedures are identified (where relevant) in the individual Evaluation Activities for each SFR.

9.3.2.1 Evaluation Activity:

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

243 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.

244 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.

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

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

247 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).

248 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.

249 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.

Findings:	The evaluator checked the requirements above are met by the guidance documentation. The 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. The CC guidance will also be published on www.niap-ccevs.org. The [AGD] following sections describe how the Operational Environment fulfil its role: - 1.3.2 Evaluated Software - 1.3.3 Non-TOE Components - 2.1 Host OS Configuration - 2.2 Configuration - 2.8 Cryptography Section 1.3.3 Non-TOE Components identifies the supported platforms for the TOE. Section 1.3.4 Evaluation Assumptions describes how the assumptions from [PP-AA] and [PP-EE] are upheld. The preparative procedures include instructions to get the drive successfully installed are provided in the [AGD]. The preparative procedures include instructions to provide a protected administrative capability in the [AGD] section Configuration.

9.4 Life-cycle Support (ALC)

9.4.1 Labelling of the TOE (ALC_CMC.1)

250 When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

Findings:	The [ST], TOE and [AGD] are all labelled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions**.**

9.4.2 TOE CM coverage (ALC_CMS.1)

251 When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

Findings:	The [ST], TOE and [AGD] are all labelled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions**.**

9.5 Tests (ATE)

9.5.1 Independent Testing – Conformance (ATE_IND.1)

252 Testing is performed to confirm the functionality described in the TSS as well as the operational guidance documentation. The focus of the testing is to confirm that the requirements specified in the SFRs are being met.

253 The evaluator should consult Appendix B FDE Equivalency Considerations when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

254 The SFR-related Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The tests identified in these other Evaluation Activities constitute a sufficient set of tests for the purposes of meeting ATE_IND.1.2E. It is important to note that while the Evaluation Activities identify the testing that is necessary to be performed, the evaluator is responsible for ensuring that the interfaces are adequately tested for the security functionality specified for each SFR.

9.5.1.1 Evaluation Activity:

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

Findings:	The TOE conforms with all configuration elements as specified in the ST.

9.5.1.2 Evaluation Activity:

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

Findings:	The evaluator verified that the TOE has been installed properly and is a known state. The evaluator followed the configuration steps found in [AGD] section 2 to ensure this was the case.

9.5.1.3 Evaluation Activity:

257 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.

Findings:	The evaluator verified that the test plan covers all of the testing actions found in ATE_IND.1 in the CEM.

258 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.

Findings:	The evaluator verified that the test plan includes and identifies the platforms that need to be tested. All firmware versions claimed in the ST are tested for all SFRs. Equivalency rationale is provided for any platforms not tested.

259 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).

Findings:	The evaluator verified the test plan describes the composition and configuration of each platform to be tested. AGD documentation was followed by the evaluator for installation and setup for each drive.

260 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.

Findings:	The evaluator verified the test plan identifies high-level test objectives as well as all test procedures to follow.

261 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.

Findings:	The evaluator verified the test report details activities that took place when all tests were executed.

10 Vulnerability Assessment

10.1 Vulnerability Survey (AVA_VAN.1)

10.1.1 Vulnerability Survey (AVA_VAN.1) Evaluation Activities

10.1.1.1 AVA_VAN.1-1

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

263 Evaluation Activity: The evaluator shall perform the CEM activity as specified.

264 *If the iTC specifies any tools to be used in performing this analysis in section A.3.4, the following text is also included in this cell: "The calibration of test resources specified in paragraph 1418 of the CEM applies to the tools listed in Appendix A, Section A.1.4."*

10.1.1.2 AVA_VAN.1-2

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

266 Evaluation Activity: The evaluator shall perform the CEM activity as specified.

10.1.1.3 AVA_VAN.1-3

267 The evaluator shall examine sources of information publicly available to identify potential vulnerabilities in the TOE.

268 Evaluation Activity: Replace CEM work unit with activities outlined in Appendix A, Section A.1.

10.1.1.4 AVA_VAN.1-4

269 The evaluator shall record in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.

270 Evaluation Activity: Replace the CEM work unit with the analysis activities on the list of potential vulnerabilities in Appendix A, section A.1, and documentation as specified in Appendix A, Section A.3.

10.1.1.5 AVA_VAN.1-5

271 The evaluator shall devise penetration tests, based on the independent search for potential vulnerabilities.

272 Evaluation Activity: Replace the CEM work unit with the activities specified in Appendix A, section A.2.

10.1.1.6 AVA_VAN.1-6

273 The evaluator shall produce penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include:

a) identification of the potential vulnerability the TOE is being tested for;

b) instructions to connect and setup all required test equipment as required to conduct the penetration test;

c) instructions to establish all penetration test prerequisite initial conditions;

d) instructions to stimulate the TSF;

e) instructions for observing the behaviour of the TSF;

f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;

g) instructions to conclude the test and establish the necessary post-test state for the TOE.

274 Evaluation Activity: The CEM work unit is captured in Appendix A, Section A.3; there are no substantive differences.

10.1.1.7 AVA_VAN.1-7

275 The evaluator shall conduct penetration testing.

276 Evaluation Activity: The evaluator shall perform the CEM activity as specified. See Appendix A, Section A.3 for guidance related to attack potential for confirmed flaws.

10.1.1.8 AVA_VAN.1-8

277 The evaluator shall record the actual results of the penetration tests.

278 Evaluation Activity: The evaluator shall perform the CEM activity as specified.

10.1.1.9 AVA_VAN.1-9

279 The evaluator shall report in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.

280 Evaluation Activity: Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.

10.1.1.10 AVA_VAN.1-10

281 The evaluator shall examine the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Basic attack potential.

282 Evaluation Activity: This work unit is not applicable for Type 1 and Type 2 flaws (as defined in Appendix A, Section A.1), as inclusion in this Supporting Document by the iTC makes any confirmed vulnerabilities stemming from these flaws subject to an attacker possessing a Basic attack potential. This work unit is replaced for Type 3 and Type 4 flaws by the activities defined in Appendix A, Section A.3.

10.1.1.11 AVA_VAN.1-11

283 The evaluator shall report in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each:

a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);

b) the SFR(s) not met;

c) a description;

d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual).

e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.

284 Evaluation Activity: Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.

285 Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in Appendix A, while an "outline" of the assurance activity is provided below.

Findings:	As noted above, the evaluation activities for AVA_VAN.1-1 through AVA_VAN.1-11 are performed in conjunction with the activities below.

10.1.1.12 Evaluation Activity (Documentation)

286 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.

287 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.

Findings:	The evaluator collected this information from the developer which was used to feed into the Type 1 Flaw Hypotheses search (below).

288 In addition to the activities specified by the CEM in accordance with Table 3 above, the evaluator shall perform the following activities.

10.1.1.13 Evaluation Activity

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

Findings:	The evaluator followed [SD-AA] and [SD-EE] Appendix A.1, A.2 and A.3 to perform the vulnerability analysis and documented the results in [VULN]. The following sources of public vulnerabilities were considered in formulating the specific list of flaws to be investigated by the evaluators, as well as to reference in

directing the evaluators to perform key-word searches during the evaluation of the TOE. Hypothesis sources for public vulnerabilities were:

NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search

Common Vulnerabilities and Exposures:
https://cve.mitre.org/cve/search_cve_list.html

US-CERT: http://www.kb.cert.org/vuls/html/search

Type 1 Hypothesis searches were last conducted on April 17, 2024 and included the following search terms:

KLC CipherDriveOne Kryptr

CipherDriveOne Kryptr

CipherDriveOne

Kryptr

Drive Encryption

Disk Encryption

Key destruction

Key sanitization

Opal management software

SED management software

Password caching

Key caching

BoringSSL

OpenSSL fips object module

Libgcrypt

Cryptsetup

OpenSSL

Opensc

Windows CryptoAPI

Linux Crypto API

Linux Kernel 5.15

The evaluation team determined that no residual vulnerabilities exist based on these searches that are exploitable by attackers with Basic Attack Potential.

	The [SD-AA] and [SD-EE] identifies Type 2 Hypotheses-iTC-Sourced. The TOE was not found to be vulnerable. This is documented in [VULN]. No type 3 or type 4 hypotheses were identified by the evaluation team.

NIAP TD0606

290 The NAS should be tested in AVA_VAN.1 to be certain that it is in the claimed evaluated configuration, locally managed with remote management disabled.

Findings:	The TOE does not include or make use of a NAS. This is not applicable.