DIGISTOR®

SECURE DATA STORAGE

DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1

Assurance Activity Report

Version 0.8

March 2023

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Reviewer	Description
0.1	03/15/22	Thibaut M.	C. Robichaud	Initial Draft
0.2	5/20/22	Dylan P.		Initial Draft
0.3	12/7/22	K. Steiner		Various updates; updated TSS verdicts
0.4	1/4/2023	K. Steiner		OR updates; Finalizing TSS, KMD, and AGD activities.
0.5	1/25/2023	K. Steiner F. Siddique	G. McLearn	Finalizing; Release to QA
0.6	2/3/2023	K. Steiner		Resolving QA comments
0.7	03/2/2023	K. Steiner	C. Cantlon	ECR comment updates
0.8	03/9/2023	K. Steiner	G. McLearn	Final updates

Table of Contents

1 Introduction

1 This Assurance Activity Report (AAR) documents the evaluation activities performed by Lightship Security 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	US Common Criteria Scheme (NIAP)
Evaluation Facility	Lightship Security USA 3600 O'Donnell St., Suite 2 Baltimore, MD 21224
Developer/Sponsor	DIGISTOR 1000 SE Tech Center Dr., Suite 160 Vancouver, WA 98683
TOE	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1
Security Target	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 Security Target, Version 1.7, March 2023
Protection Profile	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 v3.1R5
Evaluation Methodology	CEM v3.1R5
Supporting Documents	Supporting Document, Mandatory Technical Document, Full Drive Encryption: Encryption Engine, February 2019, Version 2.0 + Errata 20190201
Interpretations	**cPP_FDE_EE v2.0E** TD0458 – FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities

	TD0460 – FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states
	TD0464 – FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states
	TD0606 – FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE N/A—The TOE is not a NAS device.

1.3 Summary of SFRs

Table 3: List of SFRs

Requirement	Title
FCS_CKM.1(c)	Cryptographic Key Generation (Data Encryption Key)
FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
FCS_CKM_EXT.6	Cryptographic Key Destruction Types
FCS_KYC_EXT.2	Key Chaining (Recipient)
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FCS_VAL_EXT.1	Validation
FDP_DSK_EXT.1	Protection of Data on Disk
FMT_SMF.1	Specification of Management Functions
FPT_KYP_EXT.1	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
Selection based	
FCS_CKM.1(b)	Cryptographic Key Generation (Symmetric Keys)
FCS_CKM.4(b)	Cryptographic Key Destruction (TOE-Controlled Hardware)
FCS_COP.1(a)	Cryptographic Operation (Signature Verification)
FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)

Requirement	Title
FCS_COP.1(c)	Cryptographic Operation (Message Authentication)
FCS_COP.1(d)	Cryptographic Operation (Key Wrapping)
FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_RBG_EXT.1	Random Bit Generation
FPT_FUA_EXT.1	Firmware Update Authentication

1.4 Reference Documents

Table 4: List of Reference Documents

Ref	Document
[ST]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 Security Target, Version 1.7, March 2023
[AGD]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 Common Criteria Guide, Version 1.3, January 2023
[KMD]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 Key Management Description, Version 1.2, March 2023
[KMT]	Key_Management_Table.xls
[AVA]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 CPP_FDE_EE_v2.0E Vulnerability Assessment, Version 0.7, February 2023
[DTR]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 FDE Encryption Engine Test Plan, Version 0.5, March 2023 DIGISTOR TCG OPAL SSC FIPS SSD Series firmware version SCPG13.0/ECPG13.0/ECPM13.1 FDE Encryption Engine Test Plan Evidence, Version 0.5, March 2023
[ETR]	DIGISTOR TCG OPAL SSC FIPS SSD Series, firmware version SCPG13.0/ECPG13.0/ECPM13.1 Evaluation Technical Report, Version 0.6, March 2023
[PP]	collaborative Protection Profile for Full Drive Encryption - Encryption Engine, February 2019, Version 2.0 + Errata 20190201
[SD]	Supporting Document, Mandatory Technical Document, Full Drive Encryption: Encryption Engine, February 2019, Version 2.0 + Errata 20190201

2 TOE Details

2.1 Overview

3 The TOE is a solid state self-encrypting drive that provides encryption and decryption of stored user data.

2.2 Models/Platforms

Drive	Capacity	FIPS HW P/N & Version	CC/NIAP Listed HW P/N & Version	Controller	FW Version
DIGISTOR 2.5-Inch SATA SSD	128GB	DIG-SSD21286-SI	DIG-SSD21286-SI	PS3112-S12	SCPG13.0
	256GB	DIG-SSD22566-SI	DIG-SSD22566-SI		
	512GB	DIG-SSD25126-SI	DIG-SSD25126-SI		
	1024GB	DIG-SSD210006-SI	DIG-SSD210006-SI		
	2048GB	DIG-SSD220006-SI	DIG-SSD220006-SI		
DIGISTOR M.2 2280 SATA SSD	128GB	DIG-M21286-SI	DIG-M21286-SI		
	256GB	DIG-M22566-SI	DIG-M22566-SI		
	512GB	DIG-M25126-SI	DIG-M25126-SI		
	1024GB	DIG-M210006-SI	DIG-M210006-SI		
	2048GB	DIG-M220006-SI	DIG-M220006-SI		
DIGISTOR M.2 2280 NVMe SSD	256GB	DIG-M2N22566-UI	DIG-M2N22566-UI	PS5012-E12	ECPG13.0
	512GB	DIG-M2N25126-UI	DIG-M2N25126-UI		
	1024GB	DIG-M2N210006-UI	DIG-M2N210006-UI		
	2048GB	DIG-M2N220006-UI	DIG-M2N220006-UI		
DIGISTOR 2.5-Inch SATA SSD	128GB	DIG-SSD21286-SI	DIG-SSD212832	PS3112-S12	SCPG13.0
	256GB	DIG-SSD22566-SI	DIG-SSD225632		
	512GB	DIG-SSD25126-SI	DIG-SSD251232		
	1024GB	DIG-SSD210006-SI	DIG-SSD2100032		
	2048GB	DIG-SSD220006-SI	DIG-SSD2200032		
DIGISTOR M.2 2280 SATA SSD	128GB	DIG-M21286-SI	DIG-M212832		
	256GB	DIG-M22566-SI	DIG-M225632		

Drive	Capacity	FIPS HW P/N & Version	CC/NIAP Listed HW P/N & Version	Controller	FW Version
	512GB	DIG-M25126-SI	DIG-M251232		
	1024GB	DIG-M210006-SI	DIG-M2100032		
	2048GB	DIG-M220006-SI	DIG-M2200032		
DIGISTOR M.2 2280 NVMe SSD	256GB	DIG-M2N22566-UI	DIG-M2N225632	PS5012-E12	ECPG13.0
	512GB	DIG-M2N25126-UI	DIG-M2N251232		
	1024GB	DIG-M2N210006-UI	DIG-M2N2100032		
	2048GB	DIG-M2N220006-UI	DIG-M2N2200032		
DIGISTOR Ships Removable NVMe SSD	256GB	DIG-M2N22566-UI	Q80-M2N225632		
	512GB	DIG-M2N25126-UI	Q80-M2N251232		
	1024GB	DIG-M2N210006-UI	Q80-M2N2100032		
	2048GB	DIG-M2N220006-UI	Q80-M2N2200032		
	256GB	DIG-M2N22566-UI	Q80R-M2N225632		
	512GB	DIG-M2N25126-UI	Q80R-M2N251232		
	1024GB	DIG-M2N210006-UI	Q80R-M2N2100032		
	2048GB	DIG-M2N220006-UI	Q80R-M2N2200032		
DIGISTOR C Series FW M.2 2280 NVMe SSD	256GB	DIG-M2N22566-AI	DIG-M2N225633	PS5012-E12	ECPM13.1
	512GB	DIG-M2N25126-AI	DIG-M2N251233		
	1024GB	DIG-M2N210006-AI	DIG-M2N2100033		
	2048GB	DIG-M2N220006-AI	DIG-M2N2200033		
DIGISTOR Ships Removable C Series FW NVMe SSD	256GB	DIG-M2N22566-AI	Q80-M2N225633		
	512GB	DIG-M2N25126-AI	Q80-M2N251233		
	1024GB	DIG-M2N210006-AI	Q80-M2N2100033		
	2048GB	DIG-M2N220006-AI	Q80-M2N2200033		
	256GB	DIG-M2N22566-AI	Q80R-M2N225633		
	512GB	DIG-M2N25126-AI	Q80R-M2N251233		
	1024GB	DIG-M2N210006-AI	Q80R-M2N2100033		
	2048GB	DIG-M2N220006-AI	Q80R-M2N2200033		

2.2.1 Test Platform Equivalency

4 The evaluation team selected the DIG-M2N25126-UI, DIG-M2N25126-AI, DIG-SSD22566-SI, DIG-M2N22566-AI, and DIG-M21286-SI as the models under test. These models were selected to cover each firmware (SCPG13.0, ECPG13.0, and ECPM13.1) and controller version (PS3112-S12 and PS5012-E12) in the table above. Each firmware and controller version was fully tested. The remaining TOE models only differ in storage capacity and computer bus interface. The evaluator determined these differences to not impact the evaluated security relevant claims in [ST].

2.2.2 Testing Locations

5 Testing for the TOE drives listed below was performed at the US Lightship facility and remotely at the vendor site. The evaluator verified the TOE models and confirmed it was running the correct firmware prior to test execution. The test configuration for each TOE was isolated and tests were executed independently while being observed by the CCTL personnel, validation team and NIAP. Test platforms were further isolated by absence of a network connection, as no tests relied on network connectivity. All test evidence produced during the remote testing was transferred to the CCTL securely. Upon receipt, the CCTL verified the integrity of all test results.

2.2.3 TOE Test Configuration (testing environment)

6 The following test configurations were used:

TOE model	Platform	Controller	Firmware	SFRs
DIG-M2N25126-UI	Windows 10 Pro, Intel i5-9600	PS5012-E12	ECPG13.0	FCS_VAL_EXT.1 FMT_SMF.1 FPT_TUD_EXT.1
DIG-M2N25126-AI	Windows 10 Pro, Intel i5-9600	PS5012-E12	ECPM13.1	FCS_VAL_EXT.1 FMT_SMF.1 FPT_TUD_EXT.1
DIG-SSD22566-SI	Windows 10 Pro, Intel i5-9600	PS3112-S12	SCPG13.0	FCS_VAL_EXT.1 FMT_SMF.1 FPT_TUD_EXT.1
DIG-M2N22566-AI	Ubuntu 20.04.2 LTS, Intel i5-8400	PS5012-E12	ECPM13.1	FCS_CKM.4(b) FCS_CKM.1(c) FDP_DSK_EXT.1
DIG-SSD22566-SI	Ubuntu 16.04 LTS, Intel i7-3770K	PS3112-S12	SCPG13.0	FCS_CKM.4(b) FCS_CKM.1(c) FDP_DSK_EXT.1
DIG-M2N25126-UI	Ubuntu 20.04.2, Intel i3-8100	PS5012-E12	ECPG13.0	FCS_CKM.4(b) FCS_CKM.1(c) FDP_DSK_EXT.1

TOE model	Platform	Controller	Firmware	SFRs
DIG-M21286-SI	Ubuntu 16.04 LTS, Intel i3-8100	PS3112-S12	SCPG13.0	FCS_CKM.4(b) FCS_CKM.1(c) FDP_DSK_EXT.1

2.2.4 Tools used in the test environment

Tool name	Version	Description
KLC CipherDrive	v1.2.2	This tool provides GUI access to the TOE to be able to perform management functions
Phison Test Utility for SCPG drives	0.9.01.33	This tool was used to test the deletion and generation of key as well as provide dumps of the entire drive to verify evidence
Phison Test utility for ECPG & ECPM drives	1.10.01.01_FIPS_Digistor	This tool was used to test the deletion and generation of key as well as provide dumps of the entire drive to verify evidence
DLMC Tool for Trusted Update	ECPG13.0, ECPM13.1, SCPG13.0	This tool was used for updating the firmware on the TOE for trusted update tests.
HxD	2.5.0.0	This tool was used to verify binary file dumps with key contents

3 Evaluation Activities for SFRs

3.1 Cryptographic Support (FCS)

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

3.1.1.1 TSS

7 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:	[ST] Section 6.1.1 states the TOE generates a DEK using the *Change DEK* option in the GUI.

8 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:	The TOE generates the DEK. [ST] Section 6.1.1 states the TOE invokes the internal HMAC_DRBG when generating the DEK which is consistent with FCS_RBG_EXT.1.

9 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:	N/A. The TOE generates the DEK.

3.1.1.2 Operational Guidance

10 There are no AGD evaluation activities for this SFR.

3.1.1.3 KMD

11 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:	N/A. The TOE generates the DEK.

3.1.1.4 Test

12 The evaluator shall perform the following tests:

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

High-Level Test Description
The evaluator observed the current DEK on the TOE then queried the RBG to generate a new DEK. The evaluator observed that the DEK successfully changed and was 256 bits as claimed in the FCS_CKM.1(c) selection.

High-Level Test Description
Findings: PASS

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

3.1.2.1 TSS

14 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 6.1.3 describes how keys stored in volatile memory are destroyed. Table 12 in section 6.1.4 states when the keys are destroyed in column 'Destruction Timing'. Table 12 also describes how the memory locations for keys are tracked in column 'Storage'.

3.1.2.2 Operational Guidance

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

Findings:	N/A—the TOE does not rely on the Operational Environment for memory clearing.

3.1.2.3 KMD

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

Findings:	[KMT] describes each key type (DEK, KEK, and BEV), its origin (generation and/or establishment) and possible memory locations in volatile memory (storage).

17 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:	[KMT] and [ST] section 6.1.4 provide a key lifecycle including a description where key material resides, how the key material is used and how the material is destroyed. This is depicted in [ST] Table 12 in section 6.1.4 in the 'Storage' (where the key material resides), 'Usage' (how the key material is used) and 'Destruction' (how the key material is destroyed) columns. Section 6.1.3 also states that all keys in the chain (DEK, KEK and BEV) are erased from volatile memory when transitioning to a Compliant power saving state by performing a single overwrite of zeros. For keys in non-volatile memory, the DEK is erased by overwriting the old key with the new key then storing it in a new location in memory. The block where the old key previously resided is then erased using wear-levelling. User KEKs in non-volatile memory are erased by a single overwrite of zeros. The BEV is not stored in non-volatile memory. This is consistent with FCS_CKM.4(b) which is selected by FCS_CKM_EXT.6.

3.1.2.4 Test

18 There are no test evaluation activities for this SFR.

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

3.1.3.1 TSS

19 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:	Table 12 in [ST] Section 6.1.4 provides details for when key material is no longer needed and when they should be expected to be destroyed in column 'Destruction Timing'.

3.1.3.2 Operational Guidance

20 There are no AGD evaluation activities for this SFR.

3.1.3.3 KMD

21 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:	[KMT] Provides details on the storage location for keys. [KMD] Section 3.2.2 refers to [ST] section 6.1.4 which includes Table 12. The 'Destruction Timing' column in Table 12 describes when keys are no longer needed and destroyed.

22 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] Section 3.2.1 refers to [KMT] to describe the lifecycle of all keys. [KMT] Includes the usage, storage and zeroization of all keys. [KMD] Section 3.2.1 refers to [ST] section 6.1.4 which includes Table 12. The 'Destruction Timing' column in Table 12 describes when keys are no longer needed and destroyed. The 'Destruction' column in Table 12 describes the destruction method for all keys.

3.1.3.4 Test

23 There are no test evaluation activities for this SFR.

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

3.1.4.1 TSS

24 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:	Table 12 in [ST] Section 6.1.4 provide a description of the DEK, KEK, and BEV which are destroyed 'After a power off' which is the Compliant power saving state claimed.

3.1.4.2 Operational Guidance

25 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:	[AGD] Section 2.4 states that the TOE does not support any non-compliant power saving states.

3.1.4.3 KMD

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

Findings:	[KMT] Includes information of the storage location for all keys.

27 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:	[KMD] Section 3.2.1 refers to [KMT] to describe the lifecycle of all keys. [KMT] Includes the usage, storage and zeroization of all keys. [KMD] Section 3.2.1 refers to [ST] section 6.1.4 which includes Table 12. The 'Destruction Timing' column in Table 12 describes when keys are no longer needed and destroyed. The 'Destruction' column in Table 12 describes the destruction method for all keys.

3.1.4.4 Test

28 There are no test evaluation activities for this SFR.

3.1.5 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

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

29 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 6.1.7 states that all keys are destroyed as per the methods described in FCS_CKM.4(b). Table 12 in Section 6.1.4 lists all keys in TOE's keychain which are subject to destruction and their destruction method. The evaluator confirmed the destruction methods are consistent with the FCS_CKM.4(b) claims.

3.1.5.2 Operational Guidance

30 There are no AGD evaluation activities for this SFR.

3.1.5.3 Test

31 There are no test evaluation activities for this SFR.

3.1.6 FCS_KYC_EXT.2 Key Chaining (Recipient)

3.1.6.1 TSS

32 There are no TSS evaluation activities for this SFR.

3.1.6.2 Operational Guidance

33 There are no AGD evaluation activities for this SFR.

3.1.6.3 KMD

34 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:	[KMD] Section 3.1 provides a diagram depicting the high-level key hierarchy and details of the key chain. The description maintains a chain of keys using the key wrap method that meets FCS_COP.1(d). FCS_COP.1(e) and FCS_COP.1(g) are not claimed.

35 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:	[KMD] Section 3.1 provides a diagram showing how the key chain process functions such that is doesn't expose any material that might compromise any key in the key chain. [KMT] Provides details on where all keys and keying material are stored or what it is derived from. The key hierarchy ensures that at no point the chain could be broken without a cryptographic exhaust or knowledge of the BEV and the effective strength (256-bit) of the DEK is maintained throughout the chain. The key chain and Key Management Table provide a description of the strength of keys throughout the key chain.

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

Findings:	[KMD] Section 3.1 provides a high-level key hierarchy and details of the key chain. This includes a description of the key strength of all keys, resulting in sufficient strength to protect the 256-bit DEK.

3.1.6.4 Test

37 There are no test evaluation activities for this SFR.

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

3.1.7.1 TSS

38 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:	[ST] Section 6.1.15 describes how salts are generated. Salts are generated using the DRBG described in FCS_RBG_EXT.1. No external function is used.

39 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:	[ST] Section 6.1.15 describes how nonces are uniquely generated and appended to the encrypted data. This section also states that the Logical Block Address (LBA) is used as the tweak value. Tweak values are non-negative, consecutively assigned integers. The tweak value is converted to a little-endian byte array and encrypted using AES-XTS. This is consistent with FCS_SNI_EXT.1.3. FCS_SNI_EXT.1.3 selects 'No IV'.

3.1.7.2 Operational Guidance

40 There are no AGD evaluation activities for this SFR.

3.1.7.3 KMD

41 There are no KMD evaluation activities for this SFR.

3.1.7.4 Test

42 There are no test evaluation activities for this SFR.

3.1.8 FCS_VAL_EXT.1 Validation

3.1.8.1 TSS

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

Findings:	[ST] Section 6.1.16 states the BEV authorization factor supports validation.

44 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:	N/A—multiple submasks are not used within the TOE.

45 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:	The BEV is the only authorization factor supported by the TOE. [ST] Section 6.1.16 states the BEV is required prior to accessing the TSF data after exiting a compliant power state.

3.1.8.2 Operational Guidance

46 (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 3.4 states that the TOE requires the validation of the BEV prior to accessing the TSF data after exiting a Compliant power saving state. This can be configured to a value between 1 and 20 failed attempts in the 'Settings > Configuration > Failed Logins Before Lockout' field.

47 (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 3.4 states that the TOE can be configured to a value between 1 and 20 failed attempts in the 'Settings > Configuration > Failed Logins Before Lockout' field.

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

Findings:	[AGD] Section 3.4 states that the TOE requires the validation of the BEV prior to accessing the TSF data after exiting a Compliant power saving state.

3.1.8.3 KMD

49 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:	[KMD] Section 3.2.3 refers to [ST] section 6.1.16 which provides details on the BEV validation and the method employed to limit the number of consecutively failed authorization attempts. This is a configurable value between 1 and 20 failed attempts.

50 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:	[KMD] Section 3.2.3 refers to [ST] section 6.1.16 which provides details on the key validation. [ST] 6.1.16 states that the BEV is validated as specified in FCS_COP.1(d). Per the CPP_FDE_PP application note, validation is performed inherently when the key wrap in FCS_COP.1(d) is used.

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

Findings:	[KMD] Sections 3.1.1, 3.1.2 and 3.1.3 provides details on the key handling so that the keys and submasks are not exposed.

3.1.8.4 Test

52 The evaluator shall perform the following tests:

53 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 set the limit of failed authorization attempts and provisioned a login user. The evaluator entered invalid credentials exceeding the limit previously set. This resulted in the TOE transitioning into a locked state. The evaluator also verified when a user enters invalid credentials, the user is not allowed access to the TOE.
Findings: PASS

54 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
The evaluator proceeded to enter the TOE into a Compliant power saving mode and exited the mode. The evaluator then entered correct credentials for the user and logged in. The evaluator confirmed that invalid credentials did not result in a successful login. This test was performed in conjunction with Test 1 above.
Findings: PASS

3.2 User Data Protection (FDP)

3.2.1 FDP_DSK_EXT.1 Protection of Data on Disk

3.2.1.1 TSS

55 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] Section 6.2.1 states that the first 128MB of media data on the drive (Shadow MBR data) and the disk partition tables are read only and not encrypted. Once provisioned, all other data written to disk is encrypted without user intervention using AES-XTS. This section also details the initialization activities (when first provisioning the drive) and the boot initialization process (after it is provisioned) to ensure data is encrypted.

56 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:	N/A—no cryptographic functions are provided by the Operational Environment.

57 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] Section 6.2.1 states that once provisioned, all data written to disk is encrypted without user intervention using AES-XTS. All data stored on the TOE is encrypted except for disk partition tables and the Shadow MBR data.

58 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 6.2.1 describes the initialization of the TOE and the activities when first provisioning the drive. The TSS also describes the areas of the disk are not encrypted. Multiple disk encryptions are not supported.

3.2.1.2 Operational Guidance

59 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 2.2 refers to section 3.1 which describes the configuration that must take place in order to turn on the cryptographic module and enter FIPS mode.

3.2.1.3 KMD

60 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 provides a block diagram and description of the data encryption engine, its components, and details about its implementation. Section 4 provides a diagram showing the main components and the data path between the device's host interface and the device's persistent media storing the data. The diagram shows the location of the data encryption engine.

61 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:	[ST] Section 6.2.1 is used to provide instructions for all platforms to ensure that when the user enables encryption, the product encrypts all storage. This section also describes the data flow and states that all data is encrypted except the disk partition tables.

62 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:	[ST] Section 6.2.1 is used to provide a description of the platform's boot initialization. The TSF does not allow for the transfer of user data until all self-tests have passed and the module enters a ready state. Additionally, [KMD] section 3.2.4 states that the developer provides special tools to inspect the encrypted drive both in-band and out-of-band. The user must contact the developer to utilize these tools.

3.2.1.4 Test

63 The evaluator shall perform the following tests:

64 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
The TOE is initialized by the test utility. After the TOE is initialized, the TOE determines a random 64 KB pattern. Following the retrieval of the random pattern, the TOE retrieves information on what the device's lowest and highest logical addresses are
Findings: PASS

65 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
The test utility randomly selects several logical addresses within the devices address range. After the addresses are selected, the TOE writes the random pattern found in the previous test to these addresses.

High-Level Test Description
Findings: PASS

66 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 TOE performs an erase on the address range in the previous step. After the address range is erased, the evaluator verifies that keys cannot be found.
Findings: PASS

3.3 Security management (FMT)

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

3.3.1 FMT_SMF.1 Specification of Management Functions

3.3.1.1 TSS

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

Findings:	[ST] Section 6.3.1 describes how to change the DEK using the *Change DEK* command.

69 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:	[ST] Section 6.3.1 describes that the TOE erases the DEK as per FCS_CKM.4(a).

70 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:	[ST] Section 6.3.1 describes how to initiate the TOE firmware updates.

71 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:	[ST] Item d does not select any additional functions.

3.3.1.2 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:	[AGD] Section 3.2 provides instructions to change the DEK. [AGD] Section 3.3 states the DEKs are overwritten when the Change DEK option is executed.

72 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:	The [AGD] section 2.3 states that firmware updates are received from the vendor. Once received, updates are initiated manually by authorized administrators.

73 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:	N/A—Item d) is not selected.

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

Findings:	[AGD] Section 2.3 states that key recovery is not supported by the TOE.

3.3.1.3 KMD

75 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:	N/A. Item d) is not selected in ST.

3.3.1.4 Test

76 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
The evaluator successfully changed the DEK in the using the Change DEK option in the GUI.
Findings: PASS

77 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
The evaluator updated the TOE using a test utility. The TOE was successfully updated to the proper firmware after the test utility was executed. This was performed in conjunction with FPT_TUD_EXT.1 testing.
Findings: PASS

78 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 “no other functions.” This is N/A.
Findings: N/A

3.4 Protection of the TSF (FPT)

3.4.1 FPT_KYP_EXT.1 Protection of Key and Key Material

Technical Decision: The evaluation activities were modified per TD0458.

3.4.1.1 TSS

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

Findings: [ST] Section 6.4.2 states that the keys are wrapped using the method specified in FCS_COP.1(d).

3.4.1.2 Operational Guidance

80 There are no AGD evaluation activities for this SFR.

3.4.1.3 KMD

81 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: [KMT] Describes the storage location of all keys and the protection of all keys stored in non-volatile memory.

3.4.1.4 Test

82 There are no test evaluation activities for this SFR.

3.4.2 FPT_PWR_EXT.1 Power Saving States

Technical Decision: The evaluation activities were modified per TD0460.

3.4.2.1 TSS

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

Findings:	[ST] Section 6.4.3 states the TOE only supports the D3 Compliant power saving state.

3.4.2.2 Operational Guidance

84 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:	[AGD] Section 2.4 states that the TOE does not support any non-compliant power saving states. The TOE only supports D3 power on and off, which is a compliant power saving state.

3.4.2.3 KMD

85 There are no KMD evaluation activities for this SFR.

3.4.2.4 Test

86 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 test case is covered by evidence found in FCS_CKM.4(b)
Findings: PASS

3.4.3 FPT_PWR_EXT.2 Timing of Power Saving States

3.4.3.1 TSS

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

Findings:	[ST] Section 6.4.4 states that the TOE enters a Compliant power saving state as prompted by the protected OS and by user-initiated requests.

3.4.3.2 Operational Guidance

88 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 2.4 states only D3 (power on and power off) is supported and the time it takes the TOE to fully transition into the compliant power saving state is dependent on the host platform. In the evaluated configuration, after power is removed from the TOE, it takes approximately two seconds for DRAM to completely power down.

3.4.3.3 KMD

89 There are no KMD evaluation activities for this SFR.

3.4.3.4 Test

90 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
Evidence for this test case can be found in FCS_CKM.4(b). No power states other than powered on and powered off (D3) are supported.
Findings: PASS

3.4.4 FPT_TST_EXT.1 TSF Testing

3.4.4.1 TSS

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

Findings:	[ST] Section 6.4.5 describes the known-answer self-tests for all cryptographic functions and non-cryptographic functions in Table 13.

92 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 testing the cryptographic functions, [ST] Section 6.4.5 Table 13 also describes the Boot Loader Integrity test, DRBG Health tests and the continuous RNG test for the DRBG and NDRNG.

93 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 6.4.5 Table 13 states the Firmware DRBG Health Tests run the NIST SP 800-90A Section 11.3 Health Tests.

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

Findings:	[ST] Section 6.4.5 Table 13 describes the known-answer tests for the RSA, SHA, HMAC SHA and AES algorithms which are consistent with the FCS_COP functions.

95 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:	[ST] Section 6.4.5 Table 13 also states the Boot Loader Integrity test to test the firmware's integrity.

3.4.4.2 Operational Guidance

96 There are no AGD evaluation activities for this SFR.

3.4.4.3 KMD

97 There are no KMD evaluation activities for this SFR

3.4.4.4 Test

98 There are no test evaluation activities for this SFR.

3.4.5 FPT_TUD_EXT.1 Trusted Update

3.4.5.1 TSS

99 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:	[ST] Section 6.4.6 states that Phison signs the TOE updates. Section 6.1.8 describes how Phison is the only authorized source for code signing as the primary developer of the TOE firmware. The public key is embedded in the TOE binary and the TSS describes how the TOE behaves when the signature succeeds or fails.

100 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:	N/A—the TOE performs the signature verification and does not rely on the Operation Environment for this function.

3.4.5.2 Operational Guidance

101 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:	[AGD] Section 2.3 states that the TOE updates are received from the vendor and initiated manually. Once the update is triggered, the TOE compares the hash of the public key then verifies the digital signature. If the verification of the update succeeds, the update is applied to the TOE. If the verification of the update fails, the update process is aborted and an error is displayed to the user.

3.4.5.3 KMD

102 There are no KMD evaluation activities for this SFR.

3.4.5.4 Test

103 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):

104 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
The evaluator verified the firmware on the TOE is the current claimed firmware. After the update was applied in test 2, a version verify activity was performed to show that the update was successful.
Findings: PASS

105 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 performed an update on the TOE following the operational guidance. The evaluator verified the TOE is successfully updated.
Findings: PASS

4 Evaluation Activities for Selection-Based Requirements

4.1 Cryptographic Support (FCS)

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

106 TSS

107 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 6.1.2 states that the TOE used a 256-bit AES DEK which is protected by the KEK using the wrap function.

4.1.1.1 Operational Guidance

108 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:	[AGD] Section 3.2.1 states that the key size is set to 256-bits by default and only 256-bit keys are used in the evaluated configuration.

4.1.1.2 KMD

109 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:	[KMD] Section 3 details how the symmetric key is used as part of the key chain.

110 Test

111 There are no test evaluation activities for this SFR.

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

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

112 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:	[ST] Section 6.1.4 Table 12 describes how the keys are stored in volatile memory including the wrapping functions used and how they are overwritten. Table 12 also states how the keys are introduced to memory via the 'Initialization' column. The DEK and KEK are introduced during TOE initialization and the BEV is introduced when output from the PBKDF.

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

Findings:	[ST] Section 6.1.4 Table 12 lists each type of key that is stored and identifies the memory type where they key material is stored. The TOE uses NAND for non-volatile memory and DRAM for volatile memory.

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

Findings:	Following table 12 in section 6.1.4 of [ST], the note describes how the TOE accesses both volatile memory (DRAM) and non-volatile memory (NAND). DRAM is bit-level addressable and NAND is block-level readable and writeable. Plaintext keys are not persistently stored. Protected keys are persistently stored in NAND with parity bits and are stored in a single block that is inaccessible to the host.

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

Findings:	[ST] Section 6.1.4 identifies the destruction method and storage for each key. [KMD] Section 3.2.2 describes the destruction procedure for each key that has been identified by referencing the [KMT] and [ST] sections 6.1.3, 6.1.4 and 6.1.5 which describe the different destruction methods for each key in each type of memory.

116 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:	The ST does not make use of the open assignment.

117 The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement. Upon completion of the TSS examination, the evaluator understands how all the keys (and potential copies) are destroyed.

Findings:	[ST] There are no configurations or circumstances that may not strictly conform to the key destruction requirement. The method in which all keys are destroyed are covered.

4.1.2.2 Operational Guidance

118 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.

119 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.

120 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.

121 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.

122 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.

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

Findings:	The above activities are N/A—[AGD] section 3.3 states “The TOE does not delay key destruction of keys under any circumstance.” Thus, there are no configurations or circumstances that may not strictly conform to the key destruction requirement.

4.1.2.3 Test

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

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

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

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Cause the TOE to stop the execution but not exit.

5. Cause the TOE to dump the entire memory of the TOE into a binary file.

6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.

7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

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

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

High-Level Test Description
The evaluator recorded the values of the keys before clearing it from the TOE. After the key values are recorded, the evaluator verified the keys do indeed exist on the drive. The TOE then clears the keys from its storage. After the keys are cleared, the evaluator verifies that the keys do not exist by performing a search of the whole key and partial key fragments.
Findings: PASS

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

1. Record the value of the key in the TOE subject to clearing.

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

4. Search the non-volatile memory the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.

5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for test 1 above), and if a fragment is found in the repeated test then the test fails.

High-Level Test Description
The evaluator recorded the values of the keys before clearing it from the TOE. The keys are then cleared from the TOE and a search is performed of the whole keys and partial fragments. The evaluator verifies there are no keys found after clearing.
Findings: PASS

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

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

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

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

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

High-Level Test Description
The evaluator recorded the storage location of where the keys are stored. After the location is recorded, the evaluator verified the keys do sit at those locations. The evaluator then verified the TOE clears the key. Following the clearing of the key, the evaluator verified the keys are not stored in any of the locations recorded previously.
Findings: PASS

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

129 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.

130 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 evaluation and validation scheme must be consulted to determine the required evaluation activities.

4.1.3.1 TSS

131 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:	[ST] Section 6.1.8 describes the overall flow of the signature verification. This section states that RSA 2048 with SHA-256 is used for digital signature verification with the public key used for verification embedded in the TOE binary. The integrity of the public key is checked prior to digital signature verification. The TOE does not perform any processing that is not part of digital signature algorithm. This section (item b) also states that the obfuscated public key is embedded in the TOE binary. This key is checked against the public key hash stored in OTP memory, as stated in section 6.4.1.

4.1.3.2 Operational Guidance

132 There are no AGD evaluation activities for this SFR.

4.1.3.3 KMD

133 There are no KMD evaluation activities for this SFR.

4.1.3.4 Test

134 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.

135 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.

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

137 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.

138 **ECDSA Algorithm Tests**

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

140 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.

141 **RSA Signature Algorithm Tests**

142 **Signature Verification Test**

143 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.

144 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:	The vendor uses the CAVP certificates C1356 and C1358 for RSA signature verification. These are described in [ST] Table 4.

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

4.1.4.1 TSS

145 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:	[ST] Section 6.1.8 that the SHA-256 hash function is used for digital signature verification.

4.1.4.2 Operational Guidance

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

Findings:	[AGD] Section 3.5 states that in FIPS mode, the TOE only uses SHA-256 signature algorithm and is not configurable.

4.1.4.3 KMD

147 There are no KMD evaluation activities for this SFR.

4.1.4.4 Test

148 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.

149 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.

150 **Short Messages Test Bit-oriented Mode**

151 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.

152 **Short Messages Test Byte-oriented Mode**

153 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.

154 **Selected Long Messages Test Bit-oriented Mode**

155 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.

156 **Selected Long Messages Test Byte-oriented Mode**

157 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.

158 **Pseudorandomly Generated Messages Test**

159 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:	The vendor uses the CAVP certificates C1356 and C1358 for SHA-256 hashing. These are described in [ST] Table 4.

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

4.1.5.1 TSS

160 If HMAC was selected:

161 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:	[ST] Section 6.1.9 specifies the key length, hash function used, block size and the output MAC length.

162 If CMAC was selected:

163 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:	N/A—CMAC is not selected in [ST].

4.1.5.2 Operational Guidance

164 There are no AGD evaluation activities for this SFR.

4.1.5.3 KMD

165 There are no KMD evaluation activities for this SFR.

4.1.5.4 Test

166 If HMAC was selected:

167 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.

168 If CMAC was selected:

169 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:	The vendor uses the CAVP certificates C1356 and C1358 for HMAC-SHA2-256 message authentication. These are described in [ST] Table 4.

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

4.1.6.1 TSS

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

Findings:	[ST] Section 6.1.10 includes a description of the key wrap function. The key wrap uses AES-256.

4.1.6.2 Operational Guidance

171 There are no AGD evaluation activities for this SFR.

4.1.6.3 KMD

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

Findings:	[KMD] Section 3.1.2 states that all keys are wrapped using the approved method and a description of when the key wrapping occurs.

4.1.6.4 Test

173 There are no test evaluation activities for this SFR.

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

4.1.7.1 TSS

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

Findings:	[ST] Section 6.1.11 states AES-XTS with 256-bit keys is used for encryption and decryption. The key size used is the same for encryption and decryption.

4.1.7.2 Operational Guidance

175 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:	N/A—Only one encryption mode (AES-XTS with 256-bit keys) is supported by the TOE.

4.1.7.3 KMD

176 There are no KMD evaluation activities for this SFR.

4.1.7.4 Test

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

178 **AES-CBC Tests**

179 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.

180 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.

181 **AES-CBC Known Answer Tests**

182 KAT-1 (GFSBox):

183 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.

184 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.

185 KAT-2 (KeySBox):

186 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.

187 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.

188 KAT-3 (Variable Key):

189 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.

190 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.

191 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.

192 KAT-4 (Variable Text):

193 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.

194 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.

195 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.

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

197 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 AESCBC. The resulting ciphertext values shall be compared to the results of encrypting the plaintext messages using a known good implementation.

198 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 AESCBC. The resulting plaintext values shall be compared to the results of decrypting the ciphertext messages using a known good implementation.

199 **AES-CBC Monte Carlo Tests**

200 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.

201 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.

202 # 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]

}
```

203 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.

204 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.

205 **AES-GCM Test**

206 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.

207 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.

208 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.

209 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.

210 **XTS-AES Test**

211 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.

212 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.

213 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.

214 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:	The vendor uses the CAVP certificates C1356 and C1358 for AES-XTS encryption and decryption. These are described in [ST] Table 4.

4.1.8 FCS_KDF_EXT.1 Cryptographic Key Derivation

4.1.8.1 TSS

215 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:	[ST] Section 6.1.12 describes the key derivation function. The key derivation uses PBKDF2 using HMAC-SHA-256 with 1,000 iterations resulting in a 256-bit key in accordance with SP 800-132.

4.1.8.2 Operational Guidance

216 There are no AGD evaluation activities for this SFR.

4.1.8.3 KMD

217 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:	[KMD] Section 3 describes that all keys used are derived using an approved method and a description of how and when the keys are derived.

4.1.8.4 Test

218 There are no test evaluation activities for this SFR.

4.1.9 FCS_RBG_EXT.1 Random Bit Generation

4.1.9.1 TSS

219 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:	[ST] Section 6.1.14 includes a statement about the expected amount of entropy received. The statement is consistent with the selection made in FCS_RBG_EXT.1.2 for the seeding of the DRBG. The TOE does not use a third party entropy source.

4.1.9.2 Operational Guidance

220 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:	N/A—The hardware-based DRBG is used by default and is not configurable.

4.1.9.3 KMD

221 There are no KMD evaluation activities for this SFR.

4.1.9.4 Test

222 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.

223 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).

224 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.

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

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

227 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.

228 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.

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

Findings:	The vendor uses the CAVP certificates C1356 and C1358 for random bit generation. These are described in [ST] Table 4.

4.1.10 FPT_FUA_EXT.1 Firmware Update Authentication

4.1.10.1 TSS

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

Findings:	[ST] Section 6.4.1 states that the RTU uses a SHA-256 hash of the public key to authentication firmware updates. This has is stored in one-time programmable (OTP) memory. The firmware running on the TOE exists in ROM.

4.1.10.2 Operational Guidance

231 There are no AGD evaluation activities for this SFR

4.1.10.3 KMD

232 There are no KMD evaluation activities for this SFR.

4.1.10.4 Test

233 There are no test evaluation activities for this SFR.

5 Evaluation Activities for SARs

5.1 Security Target (ASE)

5.1.1 ASE_CCL.1 Exact Conformance Actions

5.1.1.1 ASE_CCL.1.8C

234 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_EE_V2.0E. The statements of security definition are identical in the PP and the ST.

5.1.1.2 ASE_CCL.1.9C

235 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_EE_V2.0E. The statements of security objectives are identical in the PP and the ST.

5.1.1.3 ASE_CCL.1.10C

236 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_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.

5.2 Development (ADV)

5.2.1 Basic Functional Specification (ADV_FSP.1) Evaluation Activities

5.2.1.1 ADV_FSP.1-1

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

238 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.

5.2.1.2 ADV_FSP.1-2

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

240 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.

5.2.1.3 ADV_FSP.1-3

241 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.

242 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 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.

5.2.1.4 ADV_FSP.1-4

243 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.

244 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.”

245 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:	As noted above, this work unit is covered with the rest of the ADV_FSP.1 work units.

5.2.1.5 ADV_FSP.1-5

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

247 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 and 4 of this AAR.

5.2.1.6 ADV_FSP.1-6

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

249 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:	As noted above, this work unit is covered with the EAs associated with the SFRs throughout this document.

250 ADV_FSP.1-7

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

252 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:	As noted above, this work unit is covered with the EAs associated with the SFRs throughout this document.

5.2.1.7 Evaluation Activity

253 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.

254 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.

255 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 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 and 4 of this document.

5.2.1.8 Evaluation Activity

256 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 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 and 4 of this document.

5.2.1.9 Evaluation Activity

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

258 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.

259 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.

260 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 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 and 4 of this document.

5.3 Guidance Documents (AGD)

261 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.

5.3.1 Operational User Guidance (AGD_OPE.1)

262 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).

5.3.1.1 Evaluation Activity:

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

264 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.

265 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.

266 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.

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

- The operational guidance shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.
- The operational guidance shall describe how to configure the IT environments that are supported to shut down after an administratively defined period of inactivity.
- The operational guidance shall identify system “sleeping” states for all supported operating environments and for each environment, provide administrative guidance on how to disable the sleep state. As stated above, the TOE developer may be providing an integrator’s guide and “power states” may be an abstraction that SEDs provide at various levels – e.g., may simply provide a command that the Host Platform issues to manage the state of the device, and the Host Platform is responsible for providing a more sophisticated power management scheme.
- The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

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. The section Evaluated Firmware and Hardware of the [AGD] and table 1 of the [AGD] specify the TOE and Section 1.3 of the [AGD] specifies the supported OE (Non-TOE Components). The [AGD] Section 3 provides instructions for configuring cryptographic engines. The [AGD] Section 2.4 states “No methods of inactivity timeout are supported by the TOE.” The [AGD] Section 2.4 describes the power saving states and that the TOE only supports being powered on or powered off. The evaluator verified the operational guidance documentation makes it clear which security functionality is covered by the Evaluation Activities.

5.3.2 Preparative Procedures (AGD_PRE.1)

268 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.

5.3.2.1 Evaluation Activity:

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

270 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 above.

271 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.

272 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 above.

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

274 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).

275 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.

276 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 Firmware and Hardware - 1.3.4 Non-TOE Components - 2 Configuration - 3 Cryptography

	Section 1.3.4 Non-TOE Components identifies the supported platforms for the TOE. 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.

5.4 Life-cycle Support (ALC)

5.4.1 Labelling of the TOE (ALC_CMC.1)

277 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.

5.4.2 TOE CM coverage (ALC_CMS.1)

278 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.

5.5 Tests (ATE)

5.5.1 Independent Testing – Conformance (ATE_IND.1)

279 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.

280 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.

281 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.

5.5.1.1 Evaluation Activity:

282 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.

5.5.1.2 Evaluation Activity:

283 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.

5.5.1.3 Evaluation Activity:

284 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.

285 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.

286 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.

287 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.

288 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.

5.6 Vulnerability Assessment (AVA)

5.6.1 Vulnerability Survey (AVA_VAN.1) Evaluation Activities

5.6.1.1 AVA_VAN.1-1

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

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

291 *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."*

5.6.1.2 AVA_VAN.1-2

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

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

5.6.1.3 AVA_VAN.1-3

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

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

5.6.1.4 AVA_VAN.1-4

296 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.

297 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.

5.6.1.5 AVA_VAN.1-5

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

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

5.6.1.6 AVA_VAN.1-6

300 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.

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

5.6.1.7 AVA_VAN.1-7

302 The evaluator shall conduct penetration testing.

303 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.

5.6.1.8 AVA_VAN.1-8

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

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

5.6.1.9 AVA_VAN.1-9

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

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

5.6.1.10 AVA_VAN.1-10

308 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.

309 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.

5.6.1.11 AVA_VAN.1-11

310 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.

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

312 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.

5.6.1.12 Evaluation Activity (Documentation)

313 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.

314 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).

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

5.6.1.13 Evaluation Activity

316 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] Appendix A.1, A.2 and A.3 to perform the vulnerability analysis and documented the results in [AVA]. 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 February 28, 2023 and included the following search terms:

Digistor

Trusted Computing Group (TCG)

Digistor Secure SSD

Self Encryption Drive (SED)

Digistor 2.5-Inch SATA SSD

PS3112-S12

SCPG13.0

Digistor M.2 2280 SATA SSD

Digistor M.2 2280 NVMe SSD

PS5012-E12

ECPG13.0

Digistor Ships Removable NVMe SSD

Digistor C Series FW M.2 2280 NVMe SSD

Digistor Ships Removable C Series FW NVMe SSD

ECPM13.1

Drive encryption

Disk encryption

Key destruction

Key sanitization

OPAL

ARM Cortex-R5 processor

ARMv7-R microarchitecture

Phison TCG OPAL SSC SSD solutions

	The evaluation team determined that no residual vulnerabilities exist based on these searches that are exploitable by attackers with Basic Attack Potential. The [PP] identifies a single type-2 hypotheses, however this is not applicable to the TOE since it is not a Software FDE. No type 3 or type 4 hypotheses were identified by the evaluation team.