KLC Group LLC

CipherDriveOne 2.0.1

Security Target

Version 1.3

June 2024

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	3 Jun 2024	G. NICKEL	Release for certification
1.1	18 Jun 2024	G. NICKEL	Address OR
1.2	24 Jun 2024	G. NICKEL	Address OR
1.3	26 Jun 2024	G. NICKEL	Update Guidance document references

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the CipherDriveOne Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 CipherDriveOne provides user authentication and drive/system unlock software running on an endpoint, which may be a workstation or a laptop, equipped with a Self-Encrypting Drive (SED).

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	KLC Group LLC CipherDriveOne 2.0.1 Build: 3
Security Target	KLC Group LLC CipherDriveOne 2.0.1 Security Target, v1.3

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, v2.0 + Errata 20190201 (referenced within as CPP_FDE_AA)

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Source	Applicability Rationale
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	CPP_FDE_AA	Applicable
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	CPP_FDE_AA	Not Applicable – TOE is not a NAS
TD0759	FIT Technical Decision for FCS_AFA_EXT.1.1	CPP_FDE_AA	Applicable
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	CPP_FDE_AA	Applicable
TD0764	FIT Technical Decision for FCS_PCC_EXT.1	CPP_FDE_AA	Applicable

TD #	Name	Source	Applicability Rationale
TD0765	FIT Technical Decision for FMT_MOF.1	CPP_FDE_AA	Applicable
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	CPP_FDE_AA	Applicable
TD0767	FIT Technical Decision for FMT_SMF.1.1	CPP_FDE_AA	Applicable
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	CPP_FDE_AA	Applicable

1.4 Terminology

Table 3: Terminology

Term	Definition
AA	Authorization Acquisition
AES	Advanced Encryption Standard
BEV	Border Encryption Value
BIOS	Basic Input Output System
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with CBC-Message Authentication Code
CEM	Common Evaluation Methodology
CPP	Collaborative Protection Profile
DAR	Data At Rest
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EE	Encryption Engine
EEPROM	Electrically Erasable Programmable Read-Only Memory

Term	Definition
EFI	Extensible Firmware Interface
ESP	EFI System Partition
FIPS	Federal Information Processing Standards
FDE	Full Drive Encryption
FFC	Finite Field Cryptography
GCM	Galois Counter Mode
GPT	GUID Partition Table
GUID	Globally Unique Identifier
HMAC	Keyed-Hash Message Authentication Code
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITSEF	IT Security Evaluation Facility
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IV	Initialization Vector
KEK	Key Encryption Key
KLC	KLC Group LLC
KMD	Key Management Description
KRK	Key Release Key
LKRNG	Linux Kernel Random Number Generator
MBR	Master Boot Record
NIST	National Institute of Standards and Technology
Opal 2.0	Trusted Computing Group standard for SEDs.
OS	Operating System
PBKDF	Password-Based Key Derivation Function

Term	Definition
PIV-CAC	Personal Identity Verification Common Access Card
PRF	Pseudo Random Function
PXE	Preboot eXecution Environment
RBG	Random Bit Generator
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Algorithm
RSAEP	RSA Encryption Primitive
RSADP	RSA Decryption Primitive
SAR	Security Assurance Requirements
SED	Self-Encrypting Drive
SHA	Secure Hash Algorithm
SFR	Security Functional Requirements
ST	Security Target
SPD	Security Problem Definition
SPI	Serial Peripheral Interface
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Functionality
TSS	TOE Summary Specification
UEFI	Unified Extensible Firmware Interface
USB	Universal Serial Bus
XOR	Exclusive or
XTS	XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing

2 TOE Description

2.1 Type

4 The TOE that is the subject of this evaluation against the cPP (Authorization Acquisition) is a host software solution that provides an interface for the management functionality of a self-encrypting drive (SED).

2.2 Usage

5 The TOE provides pre-boot user authentication for Opal 2.0 compliant SEDs. It is designed to be used with a SED as a loosely coupled system to deliver secure Data-At-Rest (DAR) encryption.

6 The TOE is installed on a 128MB read-only Shadow MBR partition on the SED by booting from an external USB thumb drive or DVD containing the installer. After installation, the user authenticates to the TOE (via username/password and/or smartcard) which will unlock the SED drive and chain-boot to the Protected (host) OS or Hypervisor environment.

2.3 Security Functions / Logial Scope

7 The TOE provides the following security functions:

a) **Data Protection.** The TOE enables encryption of data on a storage device to protect it from unauthorized disclosure. The TOE enables the data encryption function of a SED drive by providing pre-boot user authentication and key management capabilities.

b) **Secure Key Material.** The TOE ensures key material used for storage encryption is properly generated and protected from disclosure. It also implements cryptographic key and key material destruction during transitioning to a Compliant power saving state, or when all keys and key material are no longer needed.

c) **Secure Management.** The TOE enables management of its security functions, including:

   i) forwarding requests to change the DEK to the SED,

   ii) forwarding requests to cryptographically erase the DEK to the SED,

   iii) allowing authorized users to change authorization factors or set of authorization factors used,

   iv) initiate TOE firmware/software updates,

   v) configure authorization factors.

d) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures using RSA 3072/4096 with SHA-384/SHA-512.

e) **Cryptographic Operations.** The TOE performs cryptographic operations as shown in Table 4, which includes relevant Cryptographic Algorithm Validation Program (CAVP) certificates.

Table 4: CAVP Certificates

Module	Capability	Certificate
OpenSSL 3.2.1 Cryptographic Module	AES-CBC (128, 256)	A5187
	SHA-384 SHA-512	
	HMAC-SHA-384 HMAC-SHA-512	
	CTR_DRBG	
	RSA SigVer 186-4 (3072, 4096)	
	ECDSA KeyGen/KeyVer, SigGen/SigVer (P-256, P-384)	

2.4 Physical Scope

8 The physical boundary of the TOE encompasses the KLC CipherDriveOne software (including Linux Kernel 6.6). Users download the software after purchase from KLC's web portal. Alternatively, CipherDriveOne may come preinstalled on a partner OEM Opal2 compatible SSD/HDD disk.

2.4.1 Guidance Documents

9 The TOE includes the following guidance documents:

a) KLC Group LLC, CipherDriveOne v2.0.1, KLC PBA, 1-25-2024 (PDF)

b) KLC CipherDriveOne 2.0.1 Common Criteria Guide, 1.3 (PDF)

10 Users download the guidance documents from KLC's web portal.

2.4.2 Non-TOE Components

11 The TOE operates with the following components in the evaluated configuration:

a) **SED.** The TOE supports the following Opal 2.0 compliant Self-Encrypting Drives:

i) Advantech:
- SQFFCM8V4-256GEC (256GB M.2 NVMe)
- SQFFS25V4-256GSC (256GB 2.5" SATA)

ii) Digistor:
- DIG-SSD25126-SI (512GB 2.5" SATA)
- DIG-M2N25126-UI (512GB M.2 NVMe)

iii) Seagate:
- Barracuda 515 (512GB M.2 NVMe)
- Nytro 5350H XP1920SE70015 (1.92TB 2.5"x15mm U.3 PCIe Gen4 x4 NVMe SED)

- Nytro 5350H XP1920SE70025 (1.92TB 2.5”x15mm U.3 PCIe Gen4 x4 NVMe FIPS 140-3/Common Criteria)

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

   i) Microsoft Windows 10

   ii) Microsoft Windows 11

   iii) Red Hat Enterprise Linux 8

   iv) Red Hat Enterprise Linux 9

c) **Computer Hardware**. A minimum of 4GB of RAM and 64GB of disk storage space is recommended. The TOE supports 64-bit Intel-based systems that support UEFI including:

   i) Intel Core i5-13400 (Raptor Lake)

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

2.4.3 Security Functions not included in the TOE Evaluation

12 The evaluation is limited to those security functions identified in section 2.3.

13 The following configuration has not been evaluated:

a) Use of multiple drives

3 Security Problem Definition

14 The Security Problem Definition is reproduced from the CPP_FDE_AA.

3.1 Threats

Table 5: Threats

Identifier	Description
T.UNAUTHORIZED_ DATA_ACCESS	The cPP addresses the primary threat of unauthorized disclosure of protected data stored on a storage device. If an adversary obtains a lost or stolen storage device (e.g., a storage device contained in a laptop or a portable external storage device), they may attempt to connect a targeted storage device to a host of which they have complete control and have raw access to the storage device (e.g., to specified disk sectors, to specified blocks).
T.KEYING_MATERIAL _COMPROMISE	Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. The cPP considers possession of key material of equal importance to the data itself. Threat agents may look for key material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash.
T.AUTHORIZATION_ GUESSING	Threat agents may exercise host software to repeatedly guess authorization factors, such as passwords and PINs. Successful guessing of the authorization factors may cause the TOE to release BEV or otherwise put it in a state in which it discloses protected data to unauthorized users.
T.KEYSPACE_ EXHAUST	Threat agents may perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms and/or parameters allow attackers to exhaust the key space through brute force and give them unauthorized access to the data.
T.UNAUTHORIZED_ UPDATE	Threat agents may attempt to perform an update of the product which compromises the security features of the TOE. Poorly chosen update protocols, signature generation and verification algorithms, and parameters may allow attackers to install software and/or firmware that bypasses the intended security features and provides them unauthorized access to data.

3.2 Assumptions

Table 6: Assumptions

Identifier	Description
A.INITIAL_DRIVE_ STATE	Users enable Full Drive Encryption on a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption. The cPP does not intend to include requirements to find all the areas on storage devices that potentially contain protected data. In some cases, it may not be possible - for example, data contained in “bad” sectors. While inadvertent exposure to data contained in bad sectors or un-partitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors, un-partitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE pre-authentication software) contain no protected data.
A.SECURE_STATE	Upon the completion of proper provisioning, the drive is only assumed secure when in a powered off state up until it is powered on and receives initial authorization.
A.TRUSTED_ CHANNEL	Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure. In cases in which a single product fulfils both cPPs, then the communication between the components does not extend beyond the boundary of the TOE (e.g., communication path is within the TOE boundary). In cases in which independent products satisfy the requirements of the AA and EE, the physically close proximity of the two products during their operation means that the threat agent has very little opportunity to interpose itself in the channel between the two without the user noticing and taking appropriate actions.
A.TRAINED_USER	Authorized users follow all provided user guidance, including keeping password/passphrases and external tokens securely stored separately from the storage device and/or platform.
A.PLATFORM_STATE	The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.
A.SINGLE_USE_ET	External tokens that contain authorization factors are used for no other purpose than to store the external token authorization factors.
A.POWER_DOWN	The user does not leave the platform and/or storage device unattended until all volatile memory is cleared after a power-off, so memory remnant attacks are infeasible. Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., lock screen). Users power the platform and/or storage device down or place it into a power managed state, such as a “hibernation mode”.

Identifier	Description
A.PASSWORD_ STRENGTH	Authorized administrators ensure password/passphrase authorization factors have sufficient strength and entropy to reflect the sensitivity of the data being protected.
A.PLATFORM_I&A	The product does not interfere with or change the normal platform identification and authentication functionality such as the operating system login. It may provide authorization factors to the operating system's login interface, but it will not change or degrade the functionality of the actual interface.
A.STRONG_CRYPTO	All cryptography implemented in the Operational Environment and used by the product meets the requirements listed in the cPP. This includes generation of external token authorization factors by a RBG.
A.PHYSICAL	The platform is assumed to be physically protected in its Operational Environment and not subject to physical attacks that compromise the security and/or interfere with the platform's correct operation.

3.3 Organizational Security Policies

15 None defined.

4 Security Objectives

16 The security objectives are reproduced from the CPP_FDE_AA.

Table 7: Security Objectives for the Operational Environment

Identifier	Description
OE.TRUSTED_ CHANNEL	Communication among and between product components (i.e., AA and EE) is sufficiently protected to prevent information disclosure.
OE.INITIAL_DRIVE_ STATE	The OE provides a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption.
OE.PASSPHRASE_ STRENGTH	An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to guidance from the Enterprise using the TOE.
OE.POWER_DOWN	Volatile memory is cleared after power-off so memory remnant attacks are infeasible.
OE.SINGLE_USE_ET	External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.
OE.STRONG_ ENVIRONMENT_ CRYPTO	The Operating Environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE and Appendix A.

Identifier	Description
OE.TRAINED_USERS	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.
OE.PLATFORM_ STATE	The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.
OE.PLATFORM_I&A	The Operational Environment will provide individual user identification and authentication mechanisms that operate independently of the authorization factors used by the TOE.
OE.PHYSICAL	The Operational Environment will provide a secure physical computing space such than an adversary is not able to make modifications to the environment or to the TOE itself.

5 Security Requirements

5.1 Conventions

17 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by appending parentheses that contain a letter that is unique for each iteration, e.g. (a), (b), (c) and/or with a slash (/) followed by a descriptive string for the SFR's purpose, e.g. /Server.

18 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the PP.

5.2 Extended Components Definition

19 Refer to Annex A: Extended Components Definition.

5.3 Functional Requirements

Table 8: Summary of SFRs

Requirement	Title
FCS_AFA_EXT.1	Authorization Factor Acquisition
FCS_AFA_EXT.2	Timing of Authorization Factor Acquisition
FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
FCS_CKM.4(d)	Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
FCS_KYC_EXT.1	Key Chaining (Initiator)
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FMT_MOF.1	Management of Functions Behavior
FMT_SMF.1	Specification of Management Functions

Requirement	Title
FMT_SMR.1	Security Roles
FPT_KYP_EXT.1	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States
FPT_TUD_EXT.1	Trusted Update
Selection based	
FCS_CKM.1(b)	Cryptographic Key Generation (Symmetric Keys)
FCS_COP.1(a)	Cryptographic Operation (Signature Verification)
FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
FCS_COP.1(c)	Cryptographic Operation (Keyed Hash Algorithm)
FCS_COP.1(g)	Cryptographic Operation (Key Encryption)
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning
FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)
FCS_SMC_EXT.1	Submask Combining

5.3.1 Cryptographic Support (FCS)

FCS_AFA_EXT.1 Authorization Factor Acquisition

FCS_AFA_EXT.1.1 The TSF shall accept the following authorization factors: [

- a submask derived from a password authorization factor conditioned as defined in FCS_PCC_EXT.1,
- an external Smartcard factor that is protecting a submask that is [generated by the TOE (using the RBG as specified in FCS_RBG_EXT.1)] protected using [RSA with key size [3072 bits, 4096 bits], ECC schemes using "NIST curves" of [P-256, P-384]], with user presence proved by presentation of the smartcard and [an OE defined PIN].

].

Application Note: This SFR has been modified by TD0759.

FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

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

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

FCS_CKM.1.1(b) **Refinement:** The TSF shall generate **symmetric** cryptographic keys using **a Random Bit Generator as specified in FCS_RBG_EXT.1** and specified cryptographic key sizes [128 bit, 256 bit] that meet the following: [*no standard*].

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

FCS_CKM.4.1(a) **Refinement:** The TSF **shall [erase] cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1** that meets the following: *a key destruction method specified in FCS_CKM.4(d)*.

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

FCS_CKM.4.1(d) **Refinement:** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- **For volatile memory, the destruction shall be executed by a [**
  - **single overwrite consisting of [**
    - **a pseudo-random pattern using the TSF's RBG,**
    - **a new value of a key,**

      ],

    ];
- **For non-volatile storage that consists of the invocation of an interface provided by the underlying platform that [**
  - **instructs the underlying platform to destroy the abstraction that represents the key]**

    ]

that meets the following: [no standard].

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

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

FCS_CKM_EXT.4.1(b) Cryptographic Key and Key Material Destruction (Power Management)

FCS_CKM_EXT.4.1(b) **Refinement:** The TSF shall destroy all **key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1.**

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

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

- **RSA Digital Signature Algorithm with a key size (modulus) of [3072-bit, 4096-bit];**
- **Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater**

]

that meet the following: [

- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 29 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA schemes**
- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes**

].

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

FCS_COP.1.1(b) **Refinement:** The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [**SHA-384, SHA-512**] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ that meet the following: *ISO/IEC 10118-3:2004*.

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

FCS_COP.1.1(c) **Refinement:** The TSF shall perform cryptographic *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [**HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key sizes [***384 bits, 512 bits***] that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

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

FCS_COP.1.1(g) **Refinement:** The TSF shall perform *key encryption and decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC] *mode* and cryptographic key sizes [256 bits] that meet the following: *AES as specified in ISO /IEC 18033-3,* [CBC as specified in ISO/IEC 10116].

FCS_KDF_EXT.1 Cryptographic Key Derivation

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

- NIST SP 800-132],

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

FCS_KYC_EXT.1 Key Chaining (Initiator)

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

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

while maintaining an effective strength of [256 bits] for symmetric keys and an effective strength of [128 bits] for asymmetric keys.

Application Note: Keys are combined per FCS_SMC_EXT.1 to maintain an effective strength of 256 bits along the key chain.

FCS_KYC_EXT.1.2 The TSF shall provide at least a [128 bit, 256 bit] BEV to [*the SED*] [

- without validation taking place].

Application Note: The TOE may be configured to provide either 128 or 256 bit BEVs. The keychain remains the same in either case.

FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

FCS_PCC_EXT.1.1 A password used by the TSF to generate a password authorization factor shall enable up to [*128*] characters in the set of {upper case characters, lower case characters, numbers, and *[“!”, “”“, “#”, “$”, “%”, “&”, “‘”, “(”, “)”, “*”, “+”, “,”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “?”, “@”, “[”, “]”, “^”, “_”, “`”, “{”, “|”, “}”, “~”, “-”]*} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[SHA-384, SHA-512], with [[*100,000*] iterations], and output cryptographic key sizes [256 bits] that meet the following: [NIST SP 800-132].

Application Note: This SFR has been modified by TD0764.

FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

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

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

- [*1*] hardware-based noise source(s),]

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

FCS_SMC_EXT.1 Submask Combining

FCS_SMC_EXT.1.1 The TSF shall combine submasks using the following method [exclusive OR (XOR)] to generate an [*intermediary key or BEV*].

Application Note: Submask combining is used for dual factor authentication and only claims the generation of intermediary keys.

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

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

FCS_SNI_EXT.1.2 The TSF shall use [no nonces].

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

- CBC: IVs shall be non-repeating and unpredictable;]].

Application Note: This SFR has been modified by TD0760.

5.3.2 Security Management (FMT)

FMT_MOF.1 Management of Functions Behavior

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

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 **Refinement:** The TSF shall be capable of performing the following management functions:

a) forwarding requests to change the DEK to the EE,

b) forwarding requests to cryptographically erase the DEK to the EE,

c) allowing authorized users to change authorization values or set of authorization values used within the supported authorization method,

d) initiate TOE firmware/software updates,

e) **[configure authorization factors, disable key recovery functionality]**.

Application Note: This SFR has been modified by TD0767.

FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles *authorized user*.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.3.3 Protection of the TSF (FPT)

FPT_KYP_EXT.1 Protection of Key and Key Material

FPT_KYP_EXT.1.1 The TSF shall [

- only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1(d), or encrypted, as specified in FCS_COP.1(g) or FCS_COP.1(e)].

Application Note: Only FCS_COP.1(g) applies to the TOE and tested functionality addressed by this SFR.

FPT_PWR_EXT.1 Power Saving States

FPT_PWR_EXT.1.1 The TSF shall define the following Compliant power saving states: [S4, G2(S5), G3]

FPT_PWR_EXT.2 Timing of Power Saving States

FPT_PWR_EXT.2.1 For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur: user-initiated request, [*[as prompted by the protected OS]*].

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 **Refinement:** The TSF shall provide *authorized users* the ability to query the current version of the TOE **[software]** ~~**software/firmware.**~~

FPT_TUD_EXT.1.2 **Refinement:** The TSF shall provide *authorized users* the ability to initiate updates to TOE **[software]** ~~**software/firmware**~~.

FPT_TUD_EXT.1.3 **Refinement:** The TSF shall verify updates to the TOE software using a digital signature **as specified in FCS_COP.1(a)** by the manufacturer prior to installing those updates.

5.4 Assurance Requirements

20 The TOE security assurance requirements are summarized in Table 9.

Table 9: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - sample
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

21 In accordance with section 6.1 of the CPP_FDE_AA, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR, **including a proprietary Key Management Description (Appendix E), and [Entropy Essay].**

6 TOE Summary Specification

22 The following sections describe how the TOE fulfils each SFR included in section 5.3.

6.1 Context

6.1.1 Core TOE Concepts

23 The following are core concepts and TOE components relevant to understanding the TSS:

a) **Installer.** The TOE installer runs from a bootable device such as a USB drive, DVD or from a network share (such as executing via PXE boot). It will accept the SED administrator password and new TOE administrator password as input, bring the SED device from factory state to functional Opal state, take ownership of the SED, enable the Shadow MBR, create the ESP and install all the TOE components. At completion of the install, the hardware platform administrator sets the new TOE partition as the first boot option in the UEFI boot option list.

b) **Shadow MBR.** A 128-MB read-only partition of the SED that is the only partition visible until the SED is unlocked by the TOE. Once the SED is unlocked the Shadow MBR is mapped out and the protected partitions mapped in.

c) **ESP.** EFI System Partition (ESP) is a GPT partition with FAT32 file system located in the Shadow MBR. The system firmware loads files from this partition to boot and load the TOE.

d) **Database.** The TOE includes a database that stores the user and key tables. The database is obfuscated to prevent casual viewing and cryptographic keys are individually encrypted as described in the following sections.

e) **GUI.** The TOE provides a local GUI for PBA (SED unlock via username/password and/or smartcard) and TOE / user management.

f) **User Management.** The TOE enforces role-based access control with the following roles defined:

   i) **Admin.** Can unlock the SED, add other users and update TOE firmware.

   ii) **Security Officer.** Can unlock the SED, perform wipe-disk function and delete logs.

   iii) **Login User.** Can unlock the SED.

   iv) **Helpdesk.** Can view logs and reset user passwords.

g) **Protected OS.** The host OS or Hypervisor environment on the SED that is booted after successful TOE authentication.

6.1.2 Key Management

24 The following sections describe the fundamental key management aspects of the TOE. The figures below depict the resulting keychains designed with sufficient strength to protect a 256-bit DEK on the SED (the TOE also supports 128-bit DEKs *&* BEVs).

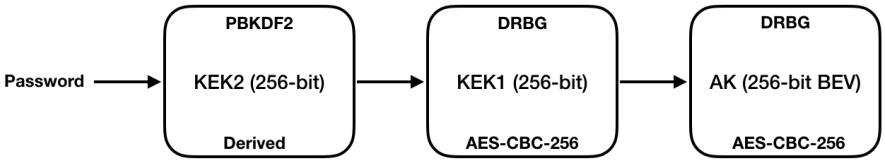

Figure 1: BEV Keychain for Password Authorization

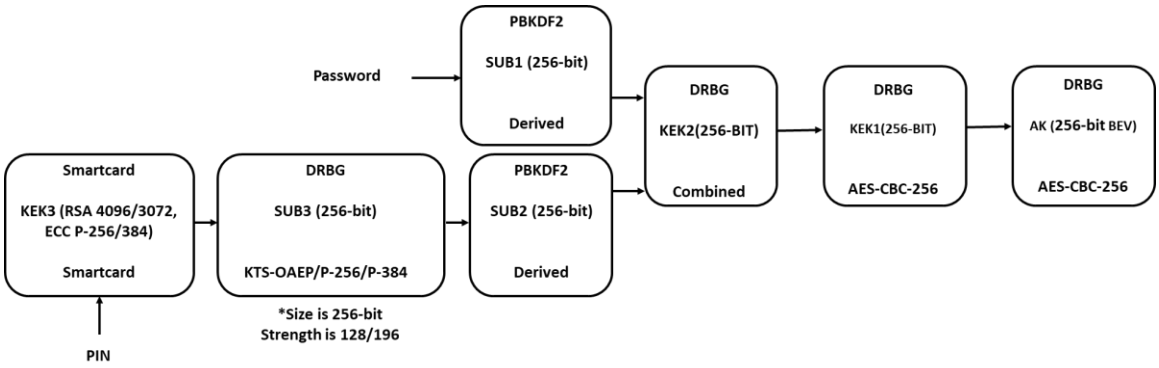

Figure 2: BEV Keychain for Dual Factor Authorization

6.1.2.1 Authentication Keys (BEV)

25 The TOE generates and manages the Authentication Keys (AKs) used to unlock a SED (AKs are the BEV referred to by the CPP_FDE_AA). The OPAL 2.0 standard specifies the following standard SED 'user accounts':

a) **SID.** Security ID – the owner of the SED (e.g. root).

b) **ADMIN SP.** This is the Administrative Security Provider. It is the OPAL construct that administers the security on the SED.

c) **LOCKING SP.** This is the Locking Security Provider. It is the OPAL construct that manages the locking and unlocking of the locking ranges on the SED.

26 During installation, the TOE generates 128-bit or 256-bit (depending on configuration AKs for the SID, ADMIN SP and LOCKING SP SED user accounts. The AKs are encrypted using AES and stored in the TOE's database.

6.1.2.2 KEKs

27 As shown in Figure 1 and Figure 2 the TOE uses a chain of up to three KEKs to the BEV (AK):

a) **KEK1.** 256-bit AES key generated by the TOE and used to encrypt the AK.

b) **KEK2.** 256-bit AES key used to encrypt KEK1. This key is derived differently depending on the authorization factors in use:

i) **Username and Password.** KEK2 is derived from the user's password via PBKDF2.

ii) **Dual Factor.** KEK2 is a combined key which is an XOR of:

- **SUB1.** 256-bit submask derived from the user's password via PBKDF2.
- **SUB2. 256-bit submask derived from SUB3 via PBKDF2**

- **SUB3.** 128-bit strength (256-bit length) submask generated by the TOE. SUB3 is RSA/ECC P-256/384 encrypted (with KEK3 public key) or signed (per below) and stored in the TOE database.

c) **KEK3.** 3072/4096 RSA or ECC P-256/384 private key stored on a smartcard and used (by the smartcard) to encrypt/sign and generate SUB3.
Note: the RSA (KTS-OAEP) decryption of SUB3 is performed by the Smartcard and not the TOE.

28 KEKs are further delineated depending on the type of user. This detail is described in the proprietary Key Management Description (KMD).

6.1.3 Authentication / SED Unlock Flow

29 At a high-level, the basic start-up and authentication flow is as follows:

a) When the TOE starts up, the database is copied, de-obfuscated and mounted in RAM. The user enters their username and password, and, if dual factor authentication is configured, presents a smartcard and PIN.

b) Depending on the authentication method:

i) Validate username against the database

ii) For smartcard, authenticate PIN against the smartcard and pass SUB3 to the smartcard for decryption and verification.

c) The TOE derives KEK2 and decrypts KEK1

d) The TOE uses the KEK1 to decrypt the appropriate AK based on the user's role

e) Provide the AK to the SED (with relevant OPAL commands)

6.2 Cryptographic Support (FCS)

6.2.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

30 The TOE supports the use of username/password and smartcards (dual factor).

6.2.1.1 Username / Password

31 The password authentication process is as follows:

a) The user enters their username and password

b) The TOE will attempt to locate the username in the database

c) The TOE will return a generic authentication error if the username is not found

d) The TOE will compare a SHA512 hash of username + password. If there is a mismatch with value computed at enrolment, the TOE will return a generic authentication error

e) The TOE will perform PBKDF2 on the password and decrypt KEK1

f) The TOE will use KEK1 to decrypt the AK

g) The TOE will use the AK to establish a session with the SED

h) If SED session establishment fails, the TOE will return a generic authentication error. If SED session creation succeeds, the user is authenticated / authorized.

6.2.1.2 Dual Factor

32 The TOE supports an external smartcard factor that is at least the same bit-length as the DEK (256-bit) - SUB3 is 256-bits in length.

33 The TOE generates SUB3 using the RBG as specified in FCS_RBG_EXT.1 and stores it in encrypted form using RSA with key size 3072 bits, 4096 bits, or ECC schemes using NIST curves of P-256 and P-384. Alternatively, if “Sign and Verify” is set, SUB3 is a signature.

34 The dual factor authentication process is as follows:

a) The user enters their username and password

b) The TOE will attempt to locate the username in the database

c) The TOE will return a generic authentication error if the username is not found

d) The TOE will compare a SHA512 hash of username + password. If there is a mismatch with value computed at enrolment, the TOE will return a generic authentication error.

e) The TOE will generate SUB1 via PBKDF2 and the user will be prompted to present a smartcard

f) The user presents a smartcard and enters the smartcard PIN

g) The smartcard will verify the PIN, if verification fails the TOE will return a generic authentication error

h) If PIN verification succeeds, the TOE will pass the encrypted SUB3 to smartcard for decryption with KEK3 or pass the random string and signature for verification.

i) The smartcard returns the decrypted SUB3 to the TOE

j) The TOE derives SUB2 from SUB3 via PBKDF2

k) The TOE combines SUB1 and SUB2 (XOR) to form KEK2

l) The TOE decrypts KEK1 with KEK2

m) The TOE will use KEK1 to decrypt the AK

n) The TOE will use the AK to establish a session with the SED

o) If SED session establishment fails, the TOE will return a generic authentication error. If SED session creation succeeds, the user is authenticated / authorized.

6.2.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

35 The user must authenticate via password or dual factor to gain access to user data after the TOE entered a Compliant power saving state described by FPT_PWT_EXT.1 below.

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

36 The TOE generates the following 256-bit AES keys: all AKs, KEK1 and SUB3.

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

37 The TOE erases cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state with a single overwrite consisting of a pseudo-random pattern or overwrite with a new value of a key.

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

38 Details regarding how keys are managed in volatile memory are provided in the KMD.

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

39 Detail regarding timing of key destruction are provided in the KMD.

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

40 Details regarding key destruction when entering a Compliant power saving state are provided in the KMD.

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

41 The TOE performs signature verification using RSA 3072 or 4096 with SHA-512 and elliptic curve algorithms using NIST curves P-256 and P-384 for trusted updates as follows:

a) TOE updates are signed with the KLC code signing private key

b) The obfuscated public key is embedded in the TOE binary

c) When the user triggers the TOE update from the GUI, the TOE verifies the digital signature using the embedded public key

d) If the digital signature verification succeeds, the upgrade process is carried out

e) If the digital signature verification fails, the upgrade process is aborted, and an error is displayed to the user.

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

42 The TOE makes use of SHA-384 and SHA-512 for digital signature verification.

43 The TOE makes use of SHA-384 and SHA-512 for PBKDF.

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

44 The TOE implements HMAC-SHA-384 and HMAC-SHA-512 with the following characteristics respectively:

a) **Key length.** 384 bits, 512 bits.

b) **Block size.** 1024 bits.

c) **MAC length.** 384 bits, 512 bits.

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

45 The TOE performs key encryption using AES-CBC-256.

6.2.12 FCS_KDF_EXT.1 Cryptographic Key Derivation

46 Passwords are conditioned via PBKDF2 using HMAC-SHA-384 or HMAC-SHA-512 with 100,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132.

6.2.13 FCS_KYC_EXT.1 Key Chaining (Initiator)

47 The TOE key chain is described at section 6.1.2.

48 The TOE supports a BEV size of 128 or 256 bits (AK) depending on configuration.

6.2.14 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

49 The TOE implements a configurable password policy with the following options:

a) Minimum Length (8 – 128)

b) Require at least one uppercase

c) Require at least one lowercase

d) Require at least one numeric

e) Require at least one special character (“!”,“”“,“#”,“$”,“%”,“&”,“‘”,“(”,“)”,“*”,“+”,“,”,“-”,“.”,“/”,“:”,“;”,“<”,“=”,“>”,“?”,“@”,“[”,“]”,“^”,“_”,“`”,“{”,“|”,“}”,“~”,“-”)

f) History (Can repeat same password after how many times)

g) No of consecutive failed validation attempts before reboot is required

50 Passwords are conditioned via PBKDF2 using HMAC-SHA-384 or HMAC-SHA-512 with 100,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132.

6.2.15 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

51 The TOE uses a software-based random bit generator (DRBG) that complies with NIST SP 800-90A for all cryptographic operations. The DRBG is seeded with entropy sourced from Linux Kernel Random Number Generator (LKRNG) operating in a blocking mode (/dev/random). All entropy is extracted, processed, and accumulated by LKRNG from multiple software-based noise sources.

6.2.16 FCS_SMC_EXT.1 Submask Combining

52 SUB1 and SUB2 are XORed together to form intermediate key KEK2.

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

53 The TOE generates 32 byte salts using the RAND_bytes function provided by the OpenSSL module at the time of encryption which are then stored in a database for use during decryption.

54 AES-CBC initialization vectors (IV) are also generated using the RAND_bytes function provided by the OpenSSL module. These IV's are appended to the encrypted data and are non-repeating and unpredictable.

55 The TOE does not make use of nonces.

6.3 Security Management (FMT)

6.3.1 FMT_MOF.1 Management of Functions Behavior

56 The TOE does not allow any modification related to power saving states.

6.3.2 FMT_SMF.1 Specification of Management Functions

57 The TOE sends the request to the SED to change the DEK in the following manner: on user's request, Opal Gen Key command is sent to the drive using the Admin AK.

58 The TOE sends the request to the SED to cryptographically erase the DEK in the following manner: on user's request for cryptographic erase of SED, Opal Revert Tper command is sent to the drive using the Admin AK.

59 The TOE GUI may be used by the user to change their password. The TOE GUI may also be used for new smartcard enrollments with a changed PIN.

60 The TOE GUI (maintenance screen) can be used to initiate updates. Key recovery functionality (export configuration or backup database) can be disabled at install time (using '-n noexport' as one of the command-line parameters) or recovery can be administratively disabled at runtime (by setting the appropriate configuration item in the Settings Console as the Security Officer).

6.3.3 FMT_SMR.1 Security Roles

61 The TOE restricts access to authorized users.

6.4 Protection of the TSF (FPT)

6.4.1 FPT_KYP_EXT.1 Protection of Key and Key Material

62 Keys are protected as described in section 6.1.2.

6.4.2 FPT_PWR_EXT.1 Power Saving States

63 The TOE supports the following Compliant power saving states:

a) **S4.** In this state, the system appears to be off and consumes lowest power. While transitioning to this state from higher power, it may save the contents of the volatile memory to a file. When the system restarts, it will load the contents of the file for a quick boot only after KLC PBA authentication/authorization.

b) **G2(S5).** In this state, the system appears to be off and involves a complete shutdown and boot process and hence KLC PBA will be invoked for authentication/authorization.

c) **G3.** In this state, the system is completely off and it does not consume any power. The system returns to the working state only after a complete reboot and hence KLC PBA will be invoked for authentication/authorization.

6.4.3 FPT_PWR_EXT.2 Timing of Power Saving States

64 The TOE enters a Compliant power saving states as prompted by the protected OS and user-initiated requests.

6.4.4 FPT_TUD_EXT.1 Trusted Update

65 Update files are digitally signed (RSA per FCS_COP.1(a)) by KLC Group and verified prior to installation.

7 Rationale

7.1 Conformance Claim Rationale

66 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is consistent with the CPP_FDE_AA.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the CPP_FDE_AA.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the CPP_FDE_AA.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the CPP_FDE_AA. No additional requirements have been specified.

7.2 Security Objectives Rationale

67 All security objectives are drawn directly from the CPP_FDE_AA.

7.3 Security Requirements Rationale

68 All security requirements are drawn directly from the CPP_FDE_AA. No optional SFRs are included in the ST. The following selection based SFRs have been included:

a) FCS_CKM.1(b)

b) FCS_COP.1(a)

c) FCS_COP.1(b)

d) FCS_COP.1(c)

e) FCS_COP.1(g)

f) FCS_KDF_EXT.1

g) FCS_PCC_EXT.1

h) FCS_RBG_EXT.1

i) FCS_SMC_EXT.1

8 Annex A: Extended Components Definition

69 Refer to Appendix ‘C’ of CPP_FDE_AA for the complete Extended Components Definition.