CIGENT

Cigent PBA Software v1.0.6

Security Target

Version 2.5

October 2023

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Description
0.1	14 March 2022	Initial draft for client review.
0.2	8 June 2022	Addressed developer comments.
0.3	24 June 2022	Initial draft for evaluation.
1.0	19 October 2022	Addressed evaluator ORs. Modified security claims and keychain support.
1.1	7 December 2022	General updates.
1.2	16 January 2023	Added FCS_RBG and Entropy into scope.
1.3	30 January 2023	Addressed evaluator comments.
1.4	13 February 2023	Updated CAVP cert #s.
1.5	27 February 2023	Address evaluator ORs.
1.6	3 March 2023	Addressed evaluator ORs.
1.7	4 April 2023	Address OR06 / ECR Comments
1.8	6 April 2023	Address Evaluator comments
1.9	18 April 2023	Address OR07
2.0	14 June 2023	Updated TOE version and addressed evaluator ORs.
2.1	19 July 2023	Added TDs
2.2	10 August 2023	Addressed ORs.
2.3	5 September 2023	Updated TOE version, CAVP, and User Guidance refs.
2.4	26 September 2023	Addressed validator comments.
2.5	10 October 2023	Addressed validator comments.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

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

2 Cigent software 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	Cigent PBA Software v1.0.6 Build: 1.0.6.4
Security Target	Cigent PBA Software v1.0.6 Security Target, v2.5

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	Applicable?	Rationale
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	CPP_FDE_AA	Yes	N/A
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	CPP_FDE_AA	No	The TOE is not a NAS.
TD0759	FIT Technical Decision for FCS_AFA_EXT.1.1	CPP_FDE_AA	Yes	N/A
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	CPP_FDE_AA	Yes	N/A
TD0764	FIT Technical Decision for FCS_PCC_EXT.1	CPP_FDE_AA	Yes	N/A

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

1.4 Terminology

Table 3: Terminology

Term	Definition
AES	Advanced Encryption Standard
AK	Authentication Key
BEV	Border Encryption Value
BIOS	Basic Input Output System
CBC	Cipher Block Chaining
CC	Common Criteria
CEM	Common Evaluation Methodology
CPP	Collaborative Protection Profile
CPU	Central Processing Unit
DAR	Data At Rest
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
EE	Encryption Engine
EFI	Extensible Firmware Interface
ESP	EFI System Partition

Term	Definition
FAT	File Allocation Table
FDE	Full Drive Encryption
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GPT	GUID Partition Table
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HMAC	Keyed-Hash Message Authentication Code
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IV	Initialization Vector
KEK	Key Encryption Key
KMD	Key Management Description
MBR	Master Boot Record
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
Opal 2.0	Trusted Computing Group standard for SEDs.
OS	Operating System
PBA	Pre-Boot Authentication
PBKDF	Password-Based Key Derivation Function
PIN	Personal Identification Number
PIV-CAC	Personal Identity Verification Common Access Card
RAM	Random Access Memory
RBG	Random Bit Generator
RSA	Rivest Shamir Adleman Algorithm
SED	Self-Encrypting Drive

Term	Definition
SFR	Security Functional Requirements
SHA	Secure Hash Algorithm
SPD	Security Problem Definition
SPI	Serial Peripheral Interface
SSD	Solid State Drive
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
UEFI	Unified Extensible Firmware Interface
USB	Universal Serial Bus

2 TOE Description

2.1 Type

4 The TOE is software that provides pre-boot authentication (PBA) for use with 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 master boot record (MBR) partition on the SED. After installation, the user authenticates to the TOE (via username/password) which unlocks the SED and chain-boots to the protected OS environment.

2.3 Security Functions / Logical 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.

d) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures using Rivest Shamir Adleman (RSA) 4096 with Secure Hash Algorithm (SHA) 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

Capability	Key/Digest Size (Bits)	Certificate
AES-GCM – Key Encryption/Creation of IVs	256	A4388
SHA – Secure Hash Algorithm	256, 512	
HMAC – Message Authentication	512	
RSA Signature Verification	4096	

Capability	Key/Digest Size (Bits)	Certificate
CTR_DRBG - Random Bit Generation (AES)	N/A	

2.4 Physical Scope

8 The physical boundary of the TOE encompasses the Cigent PBA Software v1.0.6. The TOE runs on Ubuntu 22.04 and is provided and installed as a single software package. Users may download the software after purchase from Cigent's web portal (https://download.cigent.com/Cigent_PBA_v1.0.6.zip). Alternatively, the TOE may come preinstalled on a partner original equipment manufacturer (OEM) Opal2 compatible solid state drive (SSD).

2.4.1 Guidance Documents

9 The TOE includes the following guidance documents:

a) [MAN] Cigent PBA Installation Guide and User Manual, V21, (PDF)

b) [AGD] Cigent PBA Software v1.0.6 Common Criteria Guide, v1.2, (PDF)

10 Users can download the Installation Guide and User Manual from Cigent's web portal at https://cigent.freshdesk.com/a/solutions/articles/73000595565. The Common Criteria Guide is available to users upon request.

2.4.2 Non-TOE Components

11 The TOE operates with the following components in the environment:

a) **SED**. Opal 2.0 compliant SED. CC testing performed using the following:

- Cigent Secure SSD Advanced FIPS M.2 2280

b) **Protected OS.** The TOE supports protection of Windows 10 based Operating Systems. CC testing performed using OS's:

- Microsoft Windows 10

c) **Computer Hardware.** Intel based unified extensible firmware interface (UEFI) booted systems that supports Intel Secure Key Technology. CC testing performed using the following central processing unit (CPU):

- Intel Core i7-8550U

d) **Smartcard and reader**. When dual factor authentication is used, 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.

3 Security Problem Definition

13 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 24 guess authorization factors, such as passwords and PINs. Successful guessing of the 25 authorization factors may cause the TOE to release BEV or otherwise put it in a state in which 26 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

14 None defined.

4 Security Objectives

15 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

16 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.

17 **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

18 Extended components are defined in the CPP_FDE_AA.

5.3 Functional Requirements

Table 8: Summary of Security Functional Requirements (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
FMT_SMR.1	Security Roles

Requirement	Title
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 [2048 bits], with user presence proved by presentation of the smartcard and [an OE defined PIN]

].

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 [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 [**
    - **zeroes,**
    - **ones,**

    ],
- **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(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 [4096-bit];***

]

that meet the following: [

- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA 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-256, 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-512***] and cryptographic key sizes [***256 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 **[GCM]** mode*] and cryptographic key sizes [***256 bits***] that meet the following: [*AES as specified in ISO/IEC 18033-3, **[GCM as specified in ISO/IEC 19772]***].

FCS_KDF_EXT.1 Cryptographic Key Derivation

FCS_KDF_EXT.1.1 The TSF shall accept [a RNG generated submask as specified in FCS_RBG_EXT.1, a conditioned password submask] to derive an intermediate key, as defined in [

- NIST SP 800-108 [KDF in Counter Mode],
- 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 [not applicable] for asymmetric keys.

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

- without validation taking place].

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-512], with [[*111254*] iterations], and output cryptographic key sizes [256 bits] that meet the following: [*NIST SP 800-132*].

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 [SHA-256] to generate an [*intermediary key or BEV*].

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 [

- GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key]].

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) [no other functions]].

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)].

FPT_PWR_EXT.1 Power Saving States

FPT_PWR_EXT.1.1 The TSF shall define the following Compliant power saving states: [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, [shutdown].

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

19 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

20 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

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

6.1 Context

6.1.1 Core TOE Concepts

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

a) **Installer.** The TOE installer runs from a universal serial bus (USB) drive used to boot the TOE and perform the drive configuration. 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 EFI system partition (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 GUID partition table (GPT) partition with file allocation table (FAT) 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 stores an encrypted database in the Opal provided additional 'DataStore' section. This is an up to 64MB storage area that can be read/written using specific tcg opal commands. The database will be readable by a 'datastore' user (that only has rights to read from the DataStore area). The DB will be read out of the DataStore area and stored in a temporary in-memory file. Only an authenticated user will be able to make changes (as the changes will have to be written back to the DataStore section).

e) **GUI.** The TOE provides a local graphical user interface (GUI) for PBA (SED unlock via username/password) 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. **Login User.** Can unlock the SED.

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

6.1.2 Key Management

23 The following sections describe the fundamental key management aspects of the TOE. Figures 1, 2, and 3 below depict the resulting keychains designed with sufficient strength to protect a 256-bit data encryption key (DEK).

Figure 1: Keychain for Password-Only Authorization

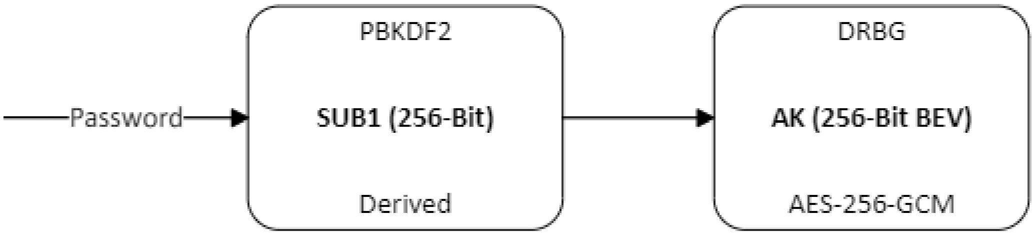


Figure 2: Keychain for Smartcard-Only Authentication

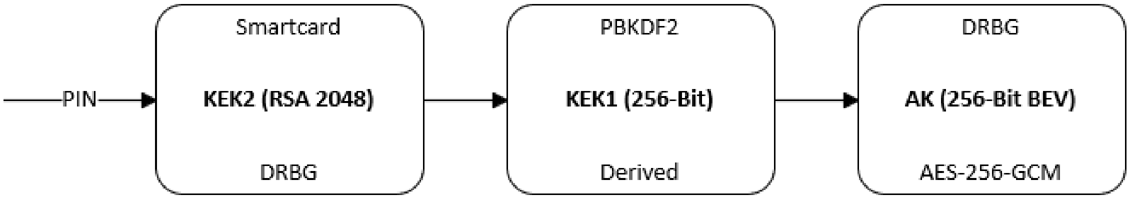
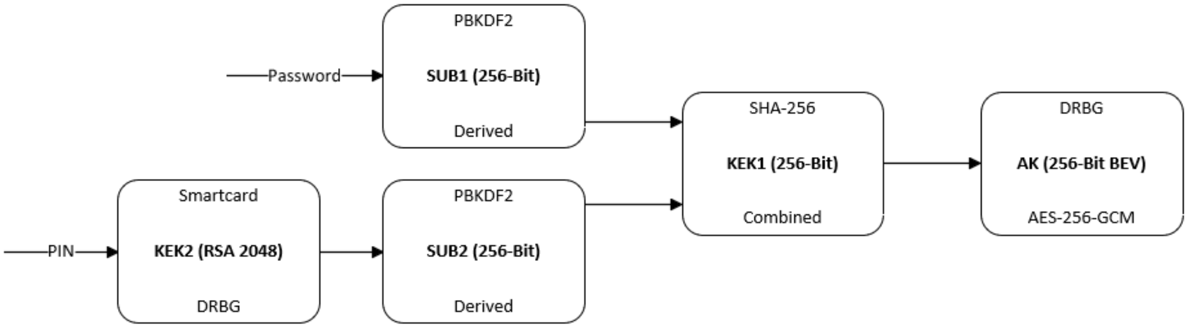


Figure 3: Keychain for Dual Factor Authorization



6.1.2.1 Authentication Keys

24 The TOE generates and manages the Authentication Keys (AKs) used to unlock a SED (AKs are the border encryption value (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.

25 During installation, the TOE generates a 256-bit BEV (AK). AKs may be generated for the SID, ADMIN SP and LOCKING SP SED user accounts. The AKs are encrypted using AES-GCM-256 and stored in the TOE's database.

6.1.2.2 Key Chain

26 As show in Figures 1, 2, and 3 the TOE uses a chain of up to two key encryption keys (KEKs) to the BEV (AK), depending on the authentication mechanism:

a) **SUB1**. 256-bit submask derived from the user's password via password-based key derivation function (PBKDF2).

b) **KEK1**. This key is derived differently depending on the authentication method and authorization factors in use:

   i. **Smartcard-Only**. The TOE feeds a 256-byte string into the smartcard, where KEK1 is PBKDF2 of the encrypted smartcard output. The deterministic random bit generator (DRBG) generated output string is stored encrypted in the TOEs database for each user.

   ii. **Dual Factor**. KEK1 is combined key which is a hash of:

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

      - **SUB2**. The TOE feeds a 256-byte string into the smartcard, where SUB2 is PBKDF2 of the encrypted smartcard output. The DRBG-generated output string is stored encrypted in the TOEs database for each user.

c) **SUB2**. The TOE feeds a 256-byte string into the smartcard, where SUB2 is PBKDF2 of the encrypted smartcard output. The DRBG-generated output string is stored encrypted in the TOEs database for each user.

d) **KEK2**. RSA private key stored on a smartcard and used (by the smartcard) to encrypt the output used for generating SUB2.

6.1.3 Authentication / Drive Unlock Flow

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

a) When the TOE starts up, it boots from the PBA code partition, launches the PBA GUI, copies and de-obfuscates the database from the PBA data partition and mounts it in random access memory (RAM). The user then enters their username and password and presents a smartcard and personal identification number (PIN).

b) Depending on the authentication method:

- For password-only, the TOE validates the username against the database.
- For smartcard-only, the smartcard PIN is authenticated.
- For dual-factor, the TOE validates the username against the database, the smartcard PIN is authenticated, and the TOE passes SUB2 to the smartcard for decryption.

c) The TOE establishes an authenticated session with the opal subsystem.

d) The TOE transitions the shadow mbr off.

e) The TOE unlocks the global range.

f) The TOE initiates a soft reboot, allowing the TOE OS (that is now accessible as the shadow mbr, transitioned off, and the global range has been unlocked) to boot.

6.2 Cryptographic Support (FCS)

6.2.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

28 The TOE supports the use of password-only, smartcard-only, and dual factor, (username/password and smartcard) authentication. The TOE supports an external smartcard factor that is at least the same bit-length as the DEK (256-bit) - SUB2 is 256-bits in length.

29 The TOE generates SUB2 using the DRBG and stores it encrypted in the database.

6.2.1.1 Password-Only Authentication Flow

30 The password-only authentication flow is as follows:

a) The user enters their username and password.

b) The TOE performs PBKDF2 on the password using additional information retrieved from the drive. If there is a username/password mismatch, the TOE will return a generic authentication error.

c) SUB1 is then used to encrypt/decrypt the AK using AES-256-GCM.

d) If the unlock succeeds, the user is authenticated / authorized and the TOE sets mbrdone to true and unlocks the global range. If the drive specific retry attempts are exhausted, the system will be shut down.

6.2.1.2 Smartcard-Only Authentication Flow

31 The smartcard-only authentication flow is as follows:

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

b) The smartcard verifies the PIN, and if the verification fails, the TOE will return a generic authentication error.

c) If PIN verification succeeds, the TOE will pass the stored DRBG-generated 256-byte string to the smartcard.

d) The TOE performs PBKDF2 on the smartcard encrypted output to generate KEK1.

e) KEK1 is then used to encrypt/decrypt the AK using AES-256-GCM.

f) If the unlock succeeds, the user is authenticated / authorized and the TOE sets mbrdone to true and unlocks the global range. If the drive specific retry attempts are exhausted, the system will be shut down.

6.2.1.3 Dual-Factor Authentication Flow

32 The dual factor authentication process is as follows:

a) The user enters their username and password.

b) The TOE performs PBKDF2 on the password using additional information retrieved from the drive. If there is a username/password mismatch, the TOE will return a generic authentication error.

c) If username/password authentication is successful, the TOE will generate SUB1 via PBKDF2 and the user will be prompted to present a smartcard.

d) The user presents a smartcard and enters the smartcard PIN.

e) The smartcard verifies the PIN, and if the verification fails, the TOE will return a generic authentication error.

f) If PIN verification succeeds, the TOE will pass the stored DRBG-generated 256-byte string to the smartcard.

g) The TOE performs PBKDF2 on the smartcard encrypted output to generate SUB2.

h) The smartcard returns the decrypted SUB2 to the TOE.

i) The TOE combines SUB1 and SUB2 (SHA-256) to form KEK1. KEK1 is then used to encrypt/decrypt the AK using AES-256-GCM.

j) If the unlock succeeds, the user is authenticated / authorized and the TOE sets mbrdone to true and unlocks the global range. If the drive specific retry attempts are exhausted, the system will be shut down.

6.2.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

33 Depending on the configuration, the user must authenticate via password-only, smartcard-only, or dual-factor to gain access to user data after the TOE enters a Compliant power saving state described by FPT_PWR_EXT.1 below.

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

34 The TOE generates 256-bit AES keys for the AK (BEV). Depending on the configuration, the BEV is protected by the authentication factors described in section 6.2.1.

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

35 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 zeroes and ones as specified in FCS_CKM.4(d). Temporary keys are not tracked.

36 **Note**: The TSF (not the Operational Environment) is used to destroy keys from volatile memory.

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

37 For volatile memory, key destruction is executed by a single overwrite consisting of zeroes and ones, immediately following the operation requiring the key is completed.

38 For non-volatile memory, the TOE GUI may be used to forward requests to cryptographically erase the DEK to the encryption engine (EE) by uninstalling the TOE or erasing the entire disk. On the admin’s request, the Opal Revert Tper command is sent to the drive followed immediately by a crypto erase (format nvm).

39 Additional details regarding how keys are managed in volatile and non-volatile memory are provided in the Key Management Description (KMD).

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

40 At a high level, keys are no longer needed when power is removed from memory, when the TOE erases the disk, or when the TOE is uninstalled. All intermediate keys are destroyed after their use in the chain. For example, for password-only authentication, SUB1 is destroyed subsequent to the decryption of the AK. Additional details 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)

41 Transitioning into the compliant power saving state automatically triggers the destruction of all keys and keying material from volatile memory.

42 Additional 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)

43 The TOE performs signature verification using RSA 4096 with SHA-512 for trusted updates as follows:

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

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

c) When the user triggers the TOE update, 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.

44 No additional processing is performed.

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

45 The TOE makes use of SHA-512 for the following:

- Digital signature verification
- PBKDF

46 The TOE make use of SHA-256 for Submask Combining as defined in FCS_SMC_EXT.1.

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

47 The TOE implements HMAC-SHA-512 with the following characteristics:

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

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

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

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

48 The TOE performs key encryption using AES-GCM-256.

6.2.12 FCS_KDF_EXT.1 Cryptographic Key Derivation

49 Passwords are conditioned via PBKDF2 using HMAC-SHA-512 with 111,254 iterations, resulting in a 256-bit key in accordance with National Institute of Standards and Technology (NIST) special publication (SP) 800-132. For smartcard authentication, the TOE accepts an RNG generated submask in accordance with NIST SP 800-108.

6.2.13 FCS_KYC_EXT.1 Key Chaining (Initiator)

50 The TOE supports a BEV (AK) size of 256 bits. Additional details on the TOE key chain are provided in section 6.1.2.

6.2.14 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

51 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 of the following special characters:
("~", "!", "@", "#", "$", "^", "&", "*", "(", ")", "_", "-", "+", "=", "[", "]", ":", "<", ">", ".")

52 Passwords are conditioned via PBKDF2 using HMAC-SHA-512 with 111,254 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)

53 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 by a single hardware-based entropy/noise source from the Intel DRNG (RDRAND/RDSEED instructions). All entropy is extracted, processed, and accumulated by OpenSSL.

54 The expected amount of entropy received from the DRNG is assumed to provide a 256-bit seed with a min-entropy of 1 bit per bit (or 8 bits per byte).

6.2.16 FCS_SMC_EXT.1 Submask Combining

55 As described in section 6.1.2, SUB1 and SUB2 are combined using SHA-256 which produces KEK1, which is used to encrypt/decrypt the 256-bit BEV.

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

56 Salts and IVs are generated using the RBG as described in FCS_RBG_EXT.1.

57 The TOE does not make use of nonces.

6.3 Security Management (FMT)

6.3.1 FMT_MOF.1 Management of Functions Behavior

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

6.3.2 FMT_SMF.1 Specification of Management Functions

59 The TOE GUI may be used to forward requests to cryptographically erase the DEK to the encryption engine (EE) via the GUI by uninstalling the TOE or erasing the entire disk. On the admin’s request, the Opal Revert Tper command is sent to the drive followed immediately by a crypto erase (format nvm).

60 **Note**: Changing the DEK is the same functionality as cryptographically erasing the DEK.

61 The TOE GUI may be used to configure the authorization factors (password-only, smartcard-only, and password + smartcard).

62 TOE updates may be performed by booting the host system with a USB drive that contains the PBA OS, utility, and the updated PBA content. An admin user must authenticate and choose to install the update.

6.3.3 FMT_SMR.1 Security Roles

63 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

64 Keys are protected as described in section 6.1.2. The AK is encrypted as per FCS_COP.1(g) and stored in the TOE’s database.

6.4.2 FPT_PWR_EXT.1 Power Saving States

65 The TOE supports the following Compliant power saving states:

a) **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 PBA will be invoked for authentication/authorization.

6.4.3 FPT_PWR_EXT.2 Timing of Power Saving States

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

6.4.4 FPT_TUD_EXT.1 Trusted Update

67 Update files are digitally signed (RSA per FCS_COP.1(a)) by Cigent and verified by the TOE prior to installation. The TOE does not support automatic update credentials. Only authorized administrators may manually perform the update process.

7 Rationale

7.1 Conformance Claim Rationale

68 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

69 All security objectives are drawn directly from the CPP_FDE_AA.

7.3 Security Requirements Rationale

70 All security requirements are drawn directly from the CPP_FDE_AA. No optional SFRs are included in the ST.

71 The following selection based SFRs have been included from PP_FDE_AA:

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