SG-KMS
Security Target

Version 1.5
Oct 2, 2023

Prepared by:
Sandhiguna
Indonesia

Table of Contents

1 Introduction

This section introduces the *security target* (ST) and identifies the *target of evaluation* (TOE). It contains the conventions, a glossary, and a list of abbreviations.

The TOE is SG-KMS, a product owned by Sandhiguna. SG-KMS manages cryptographic keys throughout their lifecycle. The cycle includes key generation, distribution, installation, rotation, revocation, recovery, expiry, and disposal. SG-KMS also provides cryptographic services, including encryption, decryption, hashing, signing, verification, tokenization, and detokenization. SG-KMS manages keys at rest, keys in transit, and keys in use and is designed specifically to protect sensitive data.

The Security Target contains the following sections:

- Section 2 provides an overview of the TOE in terms of its physical and logical boundaries.
- Section 3 defines the security problems and their environment.
- Section 4 describes the security objectives.
- Section 5 specifies the *security functional requirements* (SFRs) and *security assurance requirements* (SARs).
- Section 6 summarizes the TOE by describing the security functions on how they fulfill the SFRs.
- Section 7 supplies the reasoning behind the security problem definition, security objectives, security requirements, and security functions to justify their overall completeness, consistency, and suitability.

1.1 ST Reference

ST Title	SG-KMS Security Target
ST Version	1.5

Table 1: ST Reference

1.2 TOE Reference

TOE Name	SG-KMS
TOE Version	1.0.0012-GA

Table 2: TOE Reference

1.3 Conventions

The following conventions have been applied to this document:

- *Security functional requirements* (SFRs), CC part 2 describes an approved set of operations that may be applied to the TOE. The specification of operations is defined by iteration, assignment, selection, and refinement.
  - **Iteration**

allows a component to be used more than once due to varying operations. In this ST, iteration is indicated by a number at the end of the component. For example, FCO_NRO.1(1) and FCO_NRO.1(2) indicate that there are two iterations in the FCO_NRO.1 component.

- **Selection**
  allows the selection of one or more elements from a list of parameters. Selections are indicated using bold italics and surrounded by brackets, e.g., [***selection***].
- **Assignment**
  allows the specification of the identified parameter. The assignments are indicated using bold and are surrounded by brackets, e.g., [**assignment**]. Note that the assignment within a selection would be shown in bold italic underline, e.g., [***selected-assignment***].
- **Refinement**
  allows the addition of details. Refinements are indicated using bold for additions, e.g., …**all** keys… and strike-through for deletions e.g., …~~some~~ **all** keys….

- Other sections of the ST use bold, italic, or different font (such as `Courier`) to highlight certain text.

1.4 Abbreviations

The following abbreviations are used in this document.

AES	*Advanced Encryption Standard*
API	*Application Programming Interface*
CA	*Certificate Authority*
CC	*Common Criteria*
CCEV	*Centralized Cryptographic Engine and Vault*
CM	*Configuration Management*
DEK	*Data Encryption Key*
EK	*Encryption Key*
FIPS	*Federal Information Processing Standard*
KEK	*Key Encryption Key*
KMS	*Key Management System*
LCEV	*Local Cryptographic Engine and Vault*
MAC	*Message Authentication Code*
NIST	*National Institute of Standards and Technology*

NTP	*Network Time Protocol*
QE ID	*Quoting Enclave Identity*
PPID	*Platform Provisioning Identity*
RSA	*Rivest-Shamir-Adleman (an asymmetric cryptographic protocol)*
SAR	*Security Assurance Requirement*
SFP	*Security Function Policies*
SGX	*Software Guard Extension*
SSS	*Shamir Secret Sharing*
ST	*Security Target*
TLS	*Transport Layer Security*
TOE	*Target of Evaluation*
TSF	*TOE Security Functionality*

1.5 Glossary

Term	Description
Administrator	An interactive principal that performs administrative tasks on CCEV. In layman's term: a human.
Administrator Smart Card	A smart card owned by an administrator.
Agent	A principal that represents client's application that connects to the TOE.
API	A way for two or more computer programs to communicate with each other.
Audit Log Server	A server that records security-related activities.
CCEV	A TOE module that acts as a centralized key management system for a TOE deployment.
Certificate	An electronic document or file that proves the validity of an entity.
DEK	A key that is used by the TOE to protect user data.
DRC	Disaster Recovery Center site, one of SG-KMS deployment model that serves disaster recovery site.
Dual Control	A process that requires two or more separate entities operating in concert to either protect access to sensitive information or allow execution of important functions or roles.
Emergency Smart Card	A smart card owned by a trusted person to keep a share of the backup key.
Key Disposal	A process to dispose of keys.
Key Distribution	A process to distribute keys.
KEK_0	A key that specially protects the system slot.
Key Expiry	A process to indicate the expiry of a key.
Key Exporting	A process to export keys from the TOE, with the exported keys being secured by a standard key wrapping mechanism.
Key Generation	A process to create keys.
Key Importing	A process to import keys from another trusted party.
Key Recovery	A process to put the keys back into a TOE deployment.
Key Revocation	A process to revoke keys.
Key Rotation	A process to add a new key, which is labelled as a new version, belonging to the same key id.
LCEV	A TOE module that provides cryptographic service to the client application; keys and related cryptographic objects are downloaded by and cached locally in this module.
Mutual TLS	Two-way peer authentication using TLS protocol.
SDK	A toolkit that developers use to build applications using some prebuilt components, instead of having to build each of those components themselves.

SG-KMS RestAPI	An API that is provided by the TOE to facilitate clients' applications in communicating with the TOE.
SG-KMS SDK	An SDK that is provided by the TOE to facilitate clients' applications in communicating with the TOE in popular languages, *i.e.*, `Java and .NET`.
Shamir Secret Sharing	An information-theoretically secure cryptographic primitive that splits a secret into shares whose recovery depends on an agreed number of shares being made available.
Slot	Akin to a folder, a container to manage/group keys based on business needs.
Smart Card Channel Session Key	A key derived from secure channel keys to protect the secure channel keys.
Special Agent	A non-interactive principal who acts on the CCEV only in an emergency.
System Slot	A dedicated slot that stores special security attributes.
Termination time	Agent has termination time which is one hour. Agent can maintain this case by refresh the session manually before the termination time.
TLS Handshaking	The process in Transport Layer Security kicks off a communication session.
Token	A ticket that one possesses to access information or to perform a task.
Wrapping key	A DEK used to protect external key

Table 3: Glossary

1.6 TOE Overview

The TOE, namely, SG-KMS, is a software that provides cryptographic services and key management services to secure data. The clients deploy the TOE to generate and manage their cryptographic keys, certificates, secrets, attributes, and metadata.

SG-KMS as the TOE provides separate modules for cryptographic services and key management services. The module that handles key management services is called *centralized cryptographic engine and vault* (CCEV) whereas the module that provides cryptographic services is called *local cryptographic engine and vault* (LCEV). Depending on the specific needs of the client, SG-KMS allows for the setup of either a single LCEV or several distributed LCEVs. In running TOE, it is supported by some hardware/software/firmware described below.

Non-TOE Hardware/Software/Firmware:

The TOE needs the following external hardware/software/firmware to support its functionality:

- **SGX-enabled Intel Processors** are the main requirement as the TOE runs on such processors.
- **An Audit Log server** to record administrators' activities in CCEV. The server is logically separated from the CCEV and LCEV server. It can also be deployed in a physically separated server from the CCEV server.
- **Smart cards** come in two forms, namely administrator smart card and emergency smart card. An administrator smart card stores the administrator's public certificate. An emergency smart card stores a share of the backup key according to *Shamir Secret Sharing* scheme.
- **A dedicated smart card reader** connects to the CLI.

- **Red Hat Enterprise Linux 8.4 operating system** with all security patches installed.
- **SG-KMS SDK** is optional for users. SG-KMS SDK is available in Java.
- ***Network Time Protocol*** **(NTP)** is used by the TOE for timestamping.

1.7 TOE Description

The TOE is Sandhiguna Key Management System (SG-KMS). TOE has an access control policy to ensure that only authorized users can operate it. This policy involves some security attributes such as certificate, session, password, and smart card. The CCEV manages the keys through services. These services are key generation, distribution, rotation, revocation, recovery, expiry, exporting, importing, and disposal. The CCEV also manages slots, secrets, and certificates and regulates the access and privileges of the users. An LCEV performs cryptographic services, such as encryption, decryption, sealing, unsealing, signing, verification of a signature, tokenization, detokenization, generating MAC, and verification of a MAC. To perform these services, the LCEV can use keys which are stored either inside or outside of the TOE. Both the CCEV and the LCEVs support secure communication using mutual TLS (mTLS).

The TOE groups users into agents, administrators, and a special agent. Agents access the TOE via SG-KMS SDK or by calling SG-KMS RestAPI. Administrators and a special agent interact with the TOE via a Command Line Interface (CLI). All TOE users must be identified and authenticated by the TOE before gaining access to any TOE services. The TOE supports passwords, smart cards and/or certificate authentication.

There are two types of smart cards, namely administrator smart card and emergency smart card.

For administrator smart cards, each smart card is assigned to a TOE user assuming the administrator role. The administrator smart cards are required for Critical and High privilege operations. Please refer to Table 11 for more details on its usage.

For emergency smart cards, each smart card is assigned to TOE user assuming the special agent role. The emergency smart cards are required for Emergency privilege operations. Each emergency smart card contains an SSS share. A TOE that receives at least SSS threshold number of shares shall be able to compute the complete secret. The computed secret shall be used as an to prove the authenticity of the TOE users. Please refer to Table 11 for more details on its usage.

Upon successful user authentication, an interactive session between the TOE user and TOE is initiated. The TOE terminates the interactive session after a specified period.

The TOE provides the capability to consistently interpret the security attributes of user data when shared between TOE and another trusted IT product.

The TOE allows LCEV to operate independently from CCEV, in turn, cryptographic services provided by LCEV shall be maintained even when CCEV is down.

Outside of the TOE, an Audit Log server records security-related activities done by the administrators on the CCEV. This audit log is stored in a server that is logically separated from the CCEV and LCEV. The log keeps reliable and tamper-proof audit information. The who,

what, where, and when aspects of operations constitute the record. The TOE allows any administrator to download the audit record.

1.8 TOE Architecture

1.8.1 Deployment Architecture

The TOE is designed to be deployed in an enterprise's storage system, where it can provide cryptographic and key management services. As described above, SG-KMS architecture consists of two main components. The TOE main components are the CCEV and the LCEVs. Users interact with the TOE through some interfaces. These interfaces are *command line interface* (CLI) and RestAPI. Communications among them occur over TLS 1.3.

- An administrator and a special agent interact with the TOE by using the CLI. The administrators have the following privileges:
  - Manage administrators
  - Manage slots
  - Manage keys
  - Manage LCEVs
  - Manage client's applications
  - Manage secrets
  - Manage certificates
  - Manage a CCEV
  - Manage the configuration
- Agents that utilize SG-KMS RestAPI can perform the following services:
  - Authentication services
  - Cryptographic services
  - End-to-end encryption services
  - Certificate services
  - Secret service

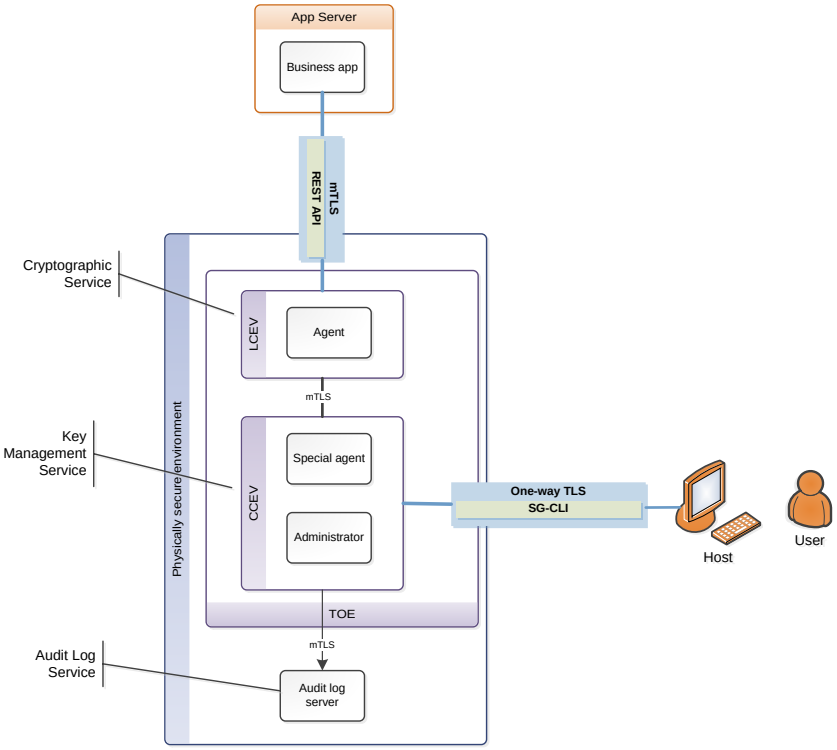

Figure 1: TOE Architecture with One LCEV

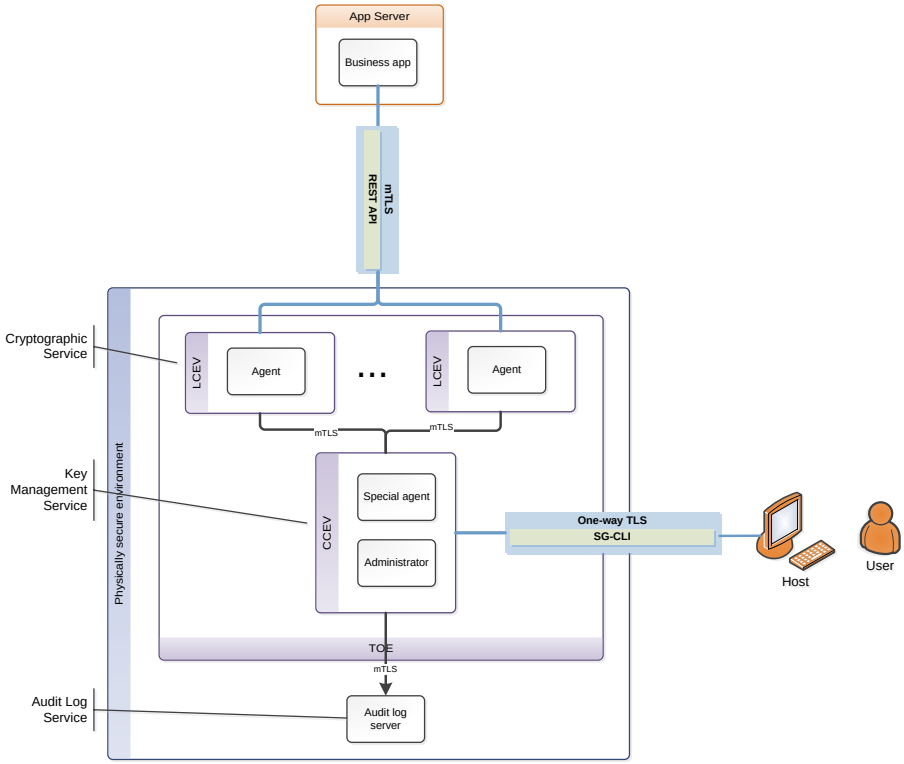

Figure 2: TOE's Architecture with several LCEVs

Figure 1: TOE Architecture with One LCEV depicts TOE's architecture with cryptographic and key management services installed in one physical server. Figure 2: TOE's Architecture with several LCEVs depicts TOE's architecture with one CCEV and some LCEVs installed in separate physical servers. In this architecture, LCEVs and the CCEV connect to each other using mTLS. The TOE is also configured with an Audit Log server, which lies outside of the TOE.

1.8.2 Software Architecture

The TOE is a software that consists of the following major components:

- ***Centralized Cryptographic Engine and Vault*** **(CCEV)** that provides key management services to the LCEVs. Their communication is secured by using mTLS.
- ***Local Cryptographic Engine and Vaults*** **(LCEVs)** that provide cryptographic services to the agents, *i.e.*, the clients' applications. Their communication is secured by using mTLS.

1.8.3 Physical Scope

The physical boundaries of the TOE are defined based on the following criteria:

- **Non-TOE Components**
  - One Audit Log Server
  - Intel SGX-enabled CPUs
  - Each CPU's Motherboard supports Intel SGX
  - Each CPU's BIOS is Intel SGX-enabled
  - Red Hat Enterprise Linux 8.4 operating system
  - Each server supports NTP
  - SG-KMS SDK
- **TOE Components**
  - One or more LCEVs
  - One CCEV
  - Deployment Guide
  - Administrator Guide

The TOE is delivered through the following methods:

Item	Filename	Form factor	Delivery
TOE	• sgkms-ccev-1.0.0012. el8.x86_64.rpm • sgkms-lcev-1.0.0012. el8.x86_64.rpm	rpm	Email with PGP protection
SG-CLI installer	• sgcli-1.0.0012. el8.x86_64.rpm	rpm	Email with PGP protection

Audit server installer	• sgkms-audit-1.0.0012.el8.x86_64.rpm	rpm	Email with PGP protection
Guidance Documents	• System Administrator Guide v1.0.0012-GA.pdf • Deployment Guide v1.0.0012-GA.pdf • SGCLI Deployment Guide v1.0.0012-GA.pdf • sgkms_api_v1.0.0012_GA.html • SG-KMS SDK	PDF/HTML	Email with PGP protection
Smart cards and readers	Not applicable	hardware	In-house delivery by registered courier

Table 4: Deliverables

1.8.4 Logical Scope

This section summarizes the security functions provided by the TOE

1.8.4.1 Cryptographic Support

The TOE provides two main services, namely, key management services and cryptographic services. The TOE is capable to perform the following cryptographic operations: symmetric encryption and decryption using AES, asymmetric encryption and decryption using RSA and ECDSA, digital signature generation and verification using RSA and ECDSA, cryptographic hashing using SHA-2, message authentication using HMAC, CMAC, and GMAC, Key Agreement Scheme using ECC, key derivation function using PBKDF2, key destruction using zeroization, key distribution using DHE and ECDHE, and secret sharing scheme using *Shamir Secret Sharing*. In support of these operations, the objects are managed by key management services. The components of the TOE are connected to each other via cryptographic protocols, *i.e.*, mTLS.

1.8.4.2 User Data Protection

User data is only distributed among clients' applications and LCEVs. Communication between them is protected by mTLS. To ensure that cryptographic services are provided securely by the LCEVs, the TOE implements an access control policy.

The TOE generates security attributes, including keys and slots, in the CCEV. These keys and slots are transmitted to the LCEVs to carry out cryptographic operations. The TOE protects the transmission by creating a secure channel that relies on mTLS.

*1.8.4.3 **Identification and Authentication***

The users of the TOE comprise administrators, agents, and a special agent who access, manage, and configure the TOE. The TOE identifies and authenticates all users before granting them access to the TOE. The TOE identifies users through their identity (IDs) and authenticates them through a password and/or a smart card which stores public certificate. Higher privileged services require more than one administrator to authenticate. The TOE locks a user account after three consecutive failed authentication attempts within a thirty-minute period.

*1.8.4.4 **Security Management***

The TOE implements a privilege-based security management model. Each service has a well-defined privilege that requires certain security attributes for identification and authentication. The privileges fall into five levels, namely emergency, critical, high, normal, and low. The details of each privilege level are described in Table 11.

The administrator and special agent connect to the CCEV to perform key management functions.

*1.8.4.5 **TSF Protection***

The TOE provides the capability to consistently interpret security attributes of user data when shared between the TOE and client application/CLI terminal.

*1.8.4.6 **TOE access***

The TOE terminates a user's interactive session based on the session token's lifetime. In addition, an agent session terminates after an hour and the agent needs to refresh the session. Different with the agent, an administrator session terminates after 10 minutes, and the administrator needs to re-login.

*1.8.4.7 **Trusted Path***

The TOE provides a trusted path between the following entities:

- agent and TOE
- administrator/special agent and TOE

The trusted path is used in the initial authentication and user operations. Such paths are protected by TLS protocol.

*1.8.4.8 **Resource Utilization***

The TOE allows LCEV to operate independently from CCEV, in turn, cryptographic services provided by LCEV shall be maintained even when CCEV is down.

*1.8.4.9 **Security Audit***

The TOE generates and keeps an audit log for administrators' activities in the CCEV. The log records the date, time, and type of each event, the identity of each relevant administrator relative to the event, and the outcome, either success or failure represented by error codes. The integrity of the audit log is guaranteed by

the HMAC on the record of each event. The entire record is stored and managed in the Audit Log Server. Any administrator can export and archive the audit log record.

1.9 Conformance Claims

1.9.1 CC Conformance

This ST and the TOE are described based on the following CC publications:

1. Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1, Revision 5, April 2017
2. Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 5, April 2017.

1.9.2 PP Conformance

The Security Target and its TOE does not conform to any Protection Profile (PP).

1.9.3 Package Conformance

The Security Target and its TOE conforms to the Evaluation Assurance Level (EAL) 2 augmented with ALC_FLR.1 component.

2 Security Problem Definition

This section defines the security problems to be addressed by the TOE, in terms of assumptions about the intended operational environment and the threats to be countered by the TOE.

2.1 Assets

2.1.1 User data

Name	Description
Plaintext	User data originating from TOE user would like to apply a cryptographic operation on.
Data Encryption Key (DEK)	It is a user data that is used by the TOE to perform cryptographic services
External Key	It is a user data that is stored in the client application. This user data is also used by the TOE to perform cryptographic services.
User secret	Secret owned by a user and is protected by TOE
User certificate	Certificate owned by a user and is protected by TOE

2.1.2 TSF Data

Name	Description
Backup Key	The key that protects the master key in backup state.
LCEV Certificate	A certificate as LCEV credential used to verify LCEV during mTLS between: • LCEV and CCEV • LCEV and client's application
Session Token	A token to create a session
Administrator reference password	It is used as a first factor authentication for administrator

2.2 Threats

This section describes the threats to be resisted by the TOE:

T.DATA_COMP	An unauthorized user may compromise the confidentiality and integrity of user data and/or TSF data being communicated between TOE users and TOE.
T.UNAUTH_ACC	An unauthorized user tries to access the TSF data through repeated password-guessing attempts

2.3 Assumption

This section describes the assumption of the TOE:

A.COMPET_USERS	TOE users are trusted and competent.
A.PHYSICAL_ENV	The TOE is deployed in a physically secure environment.
A.SECURE_PLATFORM	The underlying OS and processor are trusted and secure.
A.SMART_CARD	The user smart card is trusted and secure.
A.SC_READER	The smart card reader is trusted and secure.
A.RELIABLE_TIME	A time server shall be deployed to provide reliable timestamp to the TOE.
A.AUDIT_SERVER	An audit server shall be deployed to store the audit logs generated by the TOE. The server is trusted and secure.

2.4 Organization Security Policy

This section describes the OSP of the TOE:

OSP.CRYPTO	The TOE shall implement the cryptographic algorithms stated in **Table 10**.
OSP.DATA_AU	The TOE shall generate evidence that can be used as a guarantee of the validity of user data at rest.
OSP.DEGRADED_OP	The TOE shall ensure that LCEV remains operational even when CCEV is non-operational.
OSP.AUDIT	The TOE shall log security-relevant events.

3 Security Objective

This section identifies the security objectives for the TOE and the security objective for the operational environment.

3.1 Security Objectives for the TOE

O.I_A	The TOE shall require all users of the TOE to be identified and authenticated before gaining access to user and/or TSF data or security management functions.
O.I_A_HANDLING	The TOE shall limit the number of failed identification and authentication attempts.
O.KEY_ACC	The TOE shall provide access control to manage DEKs, user secrets and user certificates to authorized users only
O.PASS_CONTROLS	The TOE shall check users' password against a quality metric to improve password strength.
O.PROTECT_COMMS	The TOE shall protect the confidentiality and integrity of user and/or TSF data being communicated between the TOE and TOE users.
O.SECURE_MANAGE	The TOE shall enforce access control to its security management functions.
O.SESS_TERM	The TOE shall provide a mechanism to terminate a user session.
O.CRYPTO	The TOE shall implement the cryptographic algorithms stated in **Table 10**.
O.DATA_AU	The TOE shall generate evidence that can be used as a guarantee of the validity of user data at rest.
O.DEGRADED_OP	The TOE shall ensure that LCEV remains operational even when CCEV is non-operational.
O.AUDIT	The TOE shall log security-relevant events.

3.2 Security Objectives for the Operational Environment

OE.COMPET_USERS	TOE users are trusted and competent.
OE.PHYSIAL_ENV	The TOE is deployed in a physically secured environment.
OE.IT_SUPPORT_COMP	The IT supporting components are trusted and secure.
OE.RELIABLE_TIME	A time server shall be deployed to provide reliable timestamp to the TOE.

4 IT Security Requirements

4.1 TOE Security Functional Requirements

This section specifies the Security Functional Requirement (SFRs) for the TOE. SFRs were drawn from Part 2 of the Common Criteria v3.1 Revision 5 and there is no extended SFR.

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM.1: Cryptographic key generation
	FCS_CKM.2: Cryptographic key distribution
	FCS_CKM.3: Cryptographic key access
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1: Cryptographic operation
FDP: User Data Protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute-based access control
	FDP_DAU.1: Basic data authentication
	FDP_ETC.1: Export of user data without security attributes
	FDP_ITC.2: Import of user data with security attributes
FIA: Identification and Authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.1: Timing of authentication
	FIA_UAU.2: User authentication before any action
	FIA_UAU.5: Multiple authentication mechanism
	FIA_UAU.6: Re-authenticating
	FIA_UAU.7: Protected authentication feedback
	FIA_UID.1: Timing of identification
FMT: Security Management	FMT_MSA.1: Management of security attributes
	FMT_SMF.1: Specification of management functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_TDC.1: Inter-TSF basic TSF data consistency
FRU: Resource Utilization	FRU_FLT.1: Degraded fault tolerance
FTA: TOE Access	FTA_SSL.3: TSF-initiated termination
FTP: Trusted Path/Channels	FTP_TRP.1: Trusted Path
FAU: Security Audit	FAU_GEN.1: Audit data generation

Table 5: TOE Security Functional Components

4.1.1 Cryptographic Support (FCS)

4.1.1.1 Cryptographic services

FCS_COP.1(1)	**Cryptographic Operation (authenticated encryption and decryption)**
FCS_COP.1.1	The TSF shall perform [**authenticated encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES-GCM**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS PUB 197; NIST SP 800-140C; NIST SP 800-38D**]
FCS_COP.1(14)	**Cryptographic Operation (symmetric encryption and decryption)**
FCS_COP.1.1	The TSF shall perform [**symmetric encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES-CBC**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS PUB 197; NIST SP 800-140C**]
FCS_COP.1(2)	**Cryptographic Operation (asymmetric encryption and decryption)**
FCS_COP.1.1	The TSF shall perform [**asymmetric encryption and decryption**] in accordance with a specified cryptographic algorithm [**RSA-OAEP**] and cryptographic key sizes [**2048; 3072; 4096 bits for RSA**] that meet the following: [**FIPS PUB 186-4; FIPS PUB 186-5**]
FCS_COP.1(3)	**Cryptographic Operation (digital signature generation and verification)**
FCS_COP.1.1	The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**ECDSA P-256**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS PUB 186-4; FIPS PUB 186-5; NIST SP 800-140C; FIPS PUB 186-2**]
FCS_COP.1(4)	**Cryptographic Operation (digital signature generation and verification)**
FCS_COP.1.1	The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**RSA PKCS1**] and cryptographic key sizes [**2048, 3072, and 4096**] that meet the following: [**FIPS PUB 186-4; FIPS PUB 186-5; NIST SP 800-140C; FIPS PUB 186-2**]

FCS_COP.1(5)	**Cryptographic Operation (cryptographic hashing)**
FCS_COP.1.1	The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**SHA-256**] and cryptographic key sizes [**none**] that meet the following: [**FIPS PUB 180-4; FIPS PUB 202; NIST SP 800-140C**]
FCS_COP.1(6)	**Cryptographic Operation (keyed-hash message authentication)**
FCS_COP.1.1	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS 180-3; FIPS PUB 198-1; NIST SP 800-140C**]
FCS_COP.1(7)	**Cryptographic Operation (keyed-hash message authentication)**
FCS_COP.1.1	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**AES-CMAC**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS 180-3; FIPS PUB 198-1; NIST SP 800-140C**]
FCS_COP.1(8)	**Cryptographic Operation (keyed-hash message authentication)**
FCS_COP.1.1	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**AES-GMAC**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS 180-3; FIPS PUB 198-1; NIST SP 800-140C**]
FCS_CKM.1(1)	**Cryptographic key generation (RSA)**
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RSA**] and specified cryptographic key sizes [**2048; 3072; 4096 bits**] that meet the following: [**NIST SP 800-140C**]
FCS_CKM.1(2)	**Cryptographic key generation (ECDSA)**
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**ECDSA P-256**] and specified cryptographic key sizes [**256 bits**] that meet the following: [**NIST SP 800-140C**]
FCS_CKM.4	**Cryptographic key destruction**

FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**zeroization**] that meets the following: [**FIPS 140-2**]

4.1.1.2 Key Management Services

FCS_CKM.3	**Cryptographic key access**
FCS_CKM.3.1	The TSF shall perform [**cryptographic key backup**] in accordance with a specified cryptographic key access method [**Shamir Secret Sharing protocol**] that meets the following: [**none**]
Application notes:	The TOE implementation of Shamir Secret Share reference the paper **Shamir, Adi (1979), "How to Share a Secret", Communications of the ACM, 22(11): 612-613**

4.1.1.3 LCEV and CCEV Database Protection

FCS_CKM.1(3)	**Cryptographic key generation (keyed-hash message authentication)**
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**PBKDF2**] and specified cryptographic key sizes [**256 bits**] that meet the following: [**NIST SP 800-132**]

FCS_COP.1(9)	**Cryptographic Operation (keyed-hash message authentication)**
FCS_COP.1.1	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256**] and cryptographic key sizes [**256 bits**] that meet the following: [**FIPS 180-3; FIPS PUB 198-1; NIST SP 800-140C**]

FCS_COP.1(13)	**Cryptographic Operation (cryptographic hashing)**
FCS_COP.1.1	The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**Bcrypt**] and cryptographic key sizes [**none**] that meet the following: [**none**]
Application notes:	The TOE implementation of Bcrypt references the paper Provos, Niels; Mazieres, David; Talan Jason Sutton 2012 (1999). "**A Future-Adaptable Password Scheme" Proceedings of 1999 USENIX Annual Technical Conference: 81-92**

4.1.1.4 mTLS and one-way TLS

FCS_COP.1(10)	**Cryptographic Operation (TLS cryptographic hashing)**

FCS_COP.1.1	The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**SHA-256; SHA-384**] and cryptographic key sizes [**none**] that meet the following: [**FIPS PUB 180-4; FIPS PUB 202; NIST SP 800-140C**]
FCS_COP.1(11)	**Cryptographic operation (TLS symmetric encryption and decryption)**
FCS_COP.1.1	The TSF shall perform [**symmetric encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES-GCM**] and cryptographic key sizes [**128; 256 bits**] that meet the following: [**FIPS PUB 197; NIST SP 800-140C; NIST SP 800-38D**]
FCS_COP.1(12)	**Cryptographic Operation (TLS signature generation and verification)**
FCS_COP.1.1	The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**RSA PKCS1**] and cryptographic key sizes [**2048, 3072, and 4096**] that meet the following: [**FIPS PUB 186-4; FIPS PUB 186-5; NIST SP 800-140C; FIPS PUB 186-2**]
FCS_CKM.1(4)	**Cryptographic key generation (TLS)**
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**HKDF**] and specified cryptographic key sizes [**256 bits**; **384 bits**] that meet the following: [**RFC5869**]
FCS_CKM.2(1)	**Cryptographic key distribution**
FCS_CKM.2.1	The TSF shall distribute cryptographic keys in accordance with specified cryptographic key distribution method [**DHE**] that meets the following: [**NIST SP 800-52r2**]
FCS_CKM.2(2)	**Cryptographic key distribution**
FCS_CKM.2.1	The TSF shall distribute cryptographic keys in accordance with specified cryptographic key distribution method [**ECDHE**] that meets the following: [**NIST SP 800-52r2**]

4.1.2 User Data Protection (FDP)

FDP_ACC.1	**Subset access control**
FDP_ACC.1.1	The TSF shall enforce the **[user data SFP]** on **[** • **Subjects: agent, administrator, special agent**

- **Objects: slots, DEKs, user certificates, user secrets, external keys**
- **Operations:**
  - **Generate MAC with DEK**
  - **Verify MAC with DEK**
  - **Seal with DEK**
  - **Unseal with DEK**
  - **Encrypt with DEK**
  - **Decrypt with DEK**
  - **Sign with DEK**
  - **Verify signature with DEK**
  - **Sign certificate with DEK**
  - **Verify signature of certificate with DEK**
  - **Tokenize with DEK**
  - **Detokenize with DEK**
  - **Compare with DEK**
  - **Translate with DEK**
  - **Generate external key/keypair**
  - **Generate external MAC**
  - **Verify external MAC**
  - **Encrypt using external key**
  - **Decrypt using external key**
  - **Tokenize using external key**
  - **Detokenize using external key**
  - **Sign using external key**
  - **Verify signature using external key**
  - **Seal using external key**
  - **Unseal using external key**
  - **Get user secrets**
  - **Show DEK's information**
  - **List slots**
  - **Show slot information**
  - **Add slot**
  - **Change slot's password**
  - **Add application to slot**
  - **Remove application from slot**
  - **Unlock an application**
  - **Delete a slot**
  - **List DEK**
  - **Show DEK's information**
  - **Add DEK**
  - **Revoke DEK**
  - **Export DEK**
  - **Rotate DEK**

- **Import DEK PKCS11**
- **Import DEK PKCS12**
- **Update DEK**
- **Delete DEK**
- **Generate CSR using DEK**
- **List user secrets**
- **Show user secret's information**
- **Add user secret**
- **Update user secret**
- **Delete secret**
- **Generate random secret**
- **List user certificates**
- **Show user certificate's information**
- **Self-sign user certificate**
- **Import user certificate**
- **Export user certificate**
- **Renew self-signed user certificate**
- **Revoke user certificate**
- **Delete user certificate**
- **Backup DEKs, user certificates, user secrets**
- **Restore DEKs, user certificates, user secrets and slots**
- **Import backup key**]

Application notes Access to specific services is controlled by the TOE. Agents that request cryptographic services must show their certificates for authentication. The access, when granted, is restricted to a session. Access to management services, upon request by administrators, is controlled by the TOE based on password and/or smartcard authentication. Such access, when granted, is restricted to a session.

FDP_ACF.1	**Security attribute-based access control**
FDP_ACF.1.1	The TSF shall enforce the [**user data SFP**] to objects based on the following: [**Subjects: agent, special agent, administrator** **Objects: slots, user certificates, user secrets, DEKs, external keys**]
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**See Table 6**]

operations	objects	agent	administrator	special agent
Generate MAC with DEK	DEK	✓		

Verify MAC with DEK	DEK	✓		
Seal with DEK	DEK	✓		
Unseal with DEK	DEK	✓		
Encrypt with DEK	DEK	✓		
Decrypt with DEK	DEK	✓		
Sign with DEK	DEK	✓		
Verify signature with DEK	DEK	✓		
Sign certificate with DEK	DEK	✓		
Verify signature of certificate with DEK	DEK	✓		
Tokenize with DEK	DEK	✓		
Detokenize with DEK	DEK	✓		
Compare with DEK	DEK	✓		
Translate with DEK	DEK	✓		
Encrypt using external key	external keys	✓		
Decrypt using external key	external keys	✓		
Tokenize using external key	external keys	✓		
Detokenize using external key	external keys	✓		
Sign using external key	external keys	✓		
Verify signature using external key	external keys	✓		
Generate external key/keypair	external keys	✓		
Generate external MAC	external keys	✓		
Verify external MAC	external keys	✓		
Get user secrets	user secrets	✓		
List slots	slot		✓	
Show slot information	slot		✓	
Add slot	slot		✓	
Change slot’s password	slot		✓	
Add an application to a slot	slot		✓	
Remove an application from a slot	slot		✓	
Unlcok an application	slot		✓	
Delete a slot	slot		✓	
List DEKs	DEK		✓	

Show DEK's information	DEK		✓	
Add DEK	DEK		✓	
Revoke DEK	DEK		✓	
Export DEK	DEK		✓	
Rotate DEK	DEK		✓	
Import DEK PKCS11	DEK		✓	
Import DEK PKCS12	DEK		✓	
Update DEK	DEK		✓	
Delete DEK	DEK		✓	
Generate CSR	DEK		✓	
List user secrets	user secrets		✓	
Show user secret's information	user secrets		✓	
Add user secret	user secrets		✓	
Update user secret	user secrets		✓	
Delete user secret	user secrets		✓	
Generate random secret	user secrets		✓	
List user certificates	user certificates		✓	
Show user certificate's information	user certificates		✓	
Self-sign user certificate	user certificates		✓	
Import user certificate	user certificates		✓	
Export user certificate	user certificates		✓	
Renew self-signed user certificate	user certificates		✓	
Revoke user certificate	user certificates		✓	
Delete user certificate	user certificates		✓	
Backup DEK, user certificate, user secrets and slots	DEK, user certificate, user secrets and slots		✓	
Restore DEK, user certificate, user secrets and slots	DEK, user certificate, user secrets and slots			✓

Table 6: Access control

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**none**].

FDP_ACF.1.4	The TSF shall explicitly deny access of subjects based on the following additional rules: [**none**].
FDP_DAU.1	**Basic data authentication**
FDP_DAU.1.1	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [**DEK, user certificate, user secret**].
FDP_DAU.1.2	The TSF shall provide [**agent**] with the ability to verify evidence of the validity of the indicated information.
Application notes	An AES-GCM encryption key protects the integrity of user data. In addition, TOE and agent mutually authenticate one another before access the data.
FDP_ETC.1	**Export of user data without security attributes - External key, DEK, user certificate**
FDP_ETC.1.1	The TSF shall enforce the [**user data SFP**] when exporting user data, controlled under the SFPs, outside of the TOE.
FDP_ETC.1.2	The TSF shall export the user data without the user data's associated security attributes.
Application notes	Any external key or DEK or user certificate can be exported from the TOE without its security attributes.
FDP_ETC.2	**Export of user data with security attributes – User Secret**
FDP_ETC.2.1	The TSF shall enforce the [**user data SFP**] when exporting user data, controlled under the SFP, outside of the TOE.
FDP_ETC.2.2	The TSF shall export the user data with the user data's associated security attributes.
FDP_ETC.2.3	The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.
FDP_ETC.2.4	The TSF shall enforce the following rules when user data is exported from the TOE: [**none**]
FDP_ITC.1	**Import of user data without security attributes - Backup**
FDP_ITC.1.1	The TSF shall enforce the [**user data SFP**] when importing user data, controlled under the SFP, from outside of the TOE
FDP_ITC.1.2	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**none**].
FDP_ITC.2	**Import of user data with security attributes – External keys, DEK and user certificate**
FDP_ITC.2.1	The TSF shall enforce the [**user data SFP**] when importing user data, controlled under the SFP, from outside of the TOE
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside of the TOE: [**none**]
Application notes	The TOE can import any external key or DEK or user certificate with its security attributes.
FDP_RIP.1	**Subset residual information protection**
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***deallocation of the resource from***] the following objects: [**user certificate, user secret**].

4.1.3 Identification and Authentication (FIA)

FIA_AFL.1	**Authentication failure handling**
FIA_AFL.1.1	The TSF shall detect when [***three***] unsuccessful authentication attempts occur related to [**user authentication**]
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been [***surpassed***], the TSF shall [**lock the user account for 30 minutes or until the user account is unlocked by another administrator**]
FIA_ATD.1(1)	**User attribute definition - Administrator**
FIA_ATD.1.1(1)	The TSF shall maintain the following list of security attributes belonging to individual users [• **Administrator ID**

	• **Administrator public key** • **Administrator reference password** • **CCEV certificate**]
FIA_ATD.1(2)	**User attribute definition - Agent**
FIA_ATD.1.1(2)	The TSF shall maintain the following list of security attributes belonging to individual users [**LCEV certificate, Client application ID**]
FIA_ATD.1(3)	**User attribute definition - Special agent**
FIA_ATD.1.1(3)	The TSF shall maintain the following list of security attributes belonging to individual users [**backup key**]
FIA_SOS.1	**Verification of secrets - CCEV**
FIA_SOS.1.1	The TSF shall provide a mechanism to verify that secrets meet [• **between 8 and 64 characters** • **at least one upper case** • **at least one lower case** • **at least one number** • **at least one special character** • **no repeated adjacent characters** • **no more than two running characters** • **does not contain username**]
FIA_UAU.1	**Timing of authentication - CCEV**
FIA_UAU.1.1	The TSF shall allow [**configuration server and get collateral**] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Application notes	Administrators able to configure a server and get collateral without authenticated by the TOE.
FIA_UAU.2	**User authentication before any action - LCEV**

FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Application notes	An agent and the TOE, in this case the LCEV, mutually authenticate one another by using their respective certificates.

FIA_UAU.5	**Multiple authentication mechanisms - CCEV**
FIA_UAU.5.1	The TSF shall provide [• **password** • **single keypair authentication + password** • **dual keypair authentication + dual password** • **SSS authentication**] to support user authentication.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to the [**see Table 10**]

Privilege	**Operation category**	**Authentication mechanism**
Highest	Emergency	SSS authentication
↓	Critical	dual keypair authentication + dual password
	High	single keypair authentication + password
Lowest	Normal	password authentication (administrator)

Table 7: Rules for multiple authentication mechanism

FIA_UAU.6	**Re-authenticating - CCEV**
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions [**privilege of requested TSF-mediated action is higher than the current one following the hierarchy of operation category stated in Table 10**]

FIA_UAU.7	**Protected authentication feedback - CCEV**
FIA_UAU.7.1	The TSF shall provide only [• **Blank for each password character entered** • **Message to request the second authentication data**

- **Success and error message**]

to the user while the authentication is in progress.

FIA_UID.1	**Timing of identification - CCEV**
FIA_UID.1.1	The TSF shall allow [**configuration server and get collateral**] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

4.1.4 Security Management (FMT)

FMT_MSA.1	**Management of security attributes**
FMT_MSA.1.1	The TSF shall enforce the [**user data SFP**] to restrict the ability to [***delete, query, modify, add, backup***] the security attributes [**LCEV certificate, slot ID, DEK ID, DEK length, DEK purpose, DEK label, DEK lifetime, user secret ID, user secret label**] to [**administrators, special agent**].

FMT_SMF.1	**Specification of Management Functions**
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [• **Manage client's application** • **Manage administrators** • **Manage slots** • **Manage DEK** • **Manage user secrets** • **Manage user certificates** • **Manage LCEVs** • **Manage CCEVs** • **Manage administrators' password**]

FMT_SMR.1	**Security roles**
FMT_SMR.1.1	The TSF shall maintain the roles [• **special agent** • **administrator** • **agent**]
FMT_SMR.1.2	The TSF shall be able to associate users with roles
FMT_MTD.1	**Management of TSF data**
FMT_MTD.1.1	The TSF shall restrict the ability to [***modify***] the [

- **administrator reference password**
- **administrator public key,**
- **LCEV certificate**
- **backup key**

] to [**administrator**].

4.1.5 TOE Access (FTA)

FTA_SSL.3(1)	**TSF-initiated termination (Agent Session)**
FTA_SSL.3.1	The TSF shall terminate an interactive session after a **[1 hour]**
FTA_SSL.3(2)	**TSF-initiated termination (Administrator Session)**
FTA_SSL.3.1	The TSF shall terminate an interactive session after a **[10 minutes]**

4.1.6 Trusted Path (FTP)

FTP_TRP.1	**Trusted Path**
FTP_TRP.1.1	The TSF shall provide a communication path between itself and [***local, remote***] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***modification, disclosure***]
FTP_TRP.1.2	The TSF shall permit [***local users, remote users***] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for [• ***Initial user authentication*** • ***Operations described in FDP_ACC.1*** • ***Operations described in FMT_SMF.1***].

4.1.7 Protection of TSF (FPT)

FPT_TDC.1	**Inter-TSF basic TSF data consistency**
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret [**security attributes of external keys, DEK, user certificate, and user secret**] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use [

- **REST API for security attributes of external keys and user secret**
- **binary for security attributes of DEK in AES; PEM security attributes for DEK RSA and ECDSA**
- **PEM for security attributes of user certificates**

] when interpreting the TSF data from another trusted IT product.

4.1.8 Resource Utilization (FRU)

FRU_FLT.1	**Degraded fault tolerance**

FRU_FLT.1.1 The TSF shall ensure the operation of [

- **Identification and authentication of agent**
- **Generate random number**
- **Generate MAC with DEK**
- **Verify MAC with DEK**
- **Seal with DEK**
- **Unseal with DEK**
- **Encrypt with DEK**
- **Decrypt with DEK**
- **Sign with DEK**
- **Verify signature with DEK**
- **Sign certificate with DEK**
- **Verify signature of certificate with DEK**
- **Tokenize with DEK**
- **Detokenize with DEK**
- **Compare with DEK**
- **Translate with DEK**
- **Generate external key/keypair**
- **Generate external MAC**
- **Verify external MAC**
- **Encrypt using external key**
- **Decrypt using external key**
- **Tokenize using external key**
- **Detokenize using external key**
- **Sign using external key**
- **Verify signature using external key**
- **Seal using external key**
- **Unseal using external key**
- **Get user secrets**

- **Show DEK's information**

]

when the following failures occur: [**CCEV non-operation**]

4.1.9 Security Audit (FAU)

FAU_GEN.1	**Audit Data Generation**
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [see Table 8: Auditable Event Definitions] level of audit; and c) [**none**]
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**none]**

Event	Level	Comments
FCS_CKM.1(1)	Minimal	TOE records the error code for success or failure
FCS_CKM.1(2)	Minimal	TOE records the error code for success or failure
FDP_ACF.1	Basic	TOE records the error code for success or failure in perform an operation on the slots, user certificates, user secrets, and DEKs
FDP_ETC.1	Basic	TOE records the error code for success or failure in exporting DEK or user certificate
FDP_ITC.2	Basic	TOE records the error code for success or failure in importing DEK or user certificate
FIA_AFL.1	Minimal	TOE records error code for authentication failure
FIA_UAU.1	Basic	TOE records error code for success or failure authentication
FIA_UAU.5	Minimal	TOE records error code for final decision on authentication

FIA_UAU.6	Basic	TOE records error code for success or failure reauthentication
FIA_UID.1	Basic	TOE records administrator identity
FMT_MSA.1	Basic	TOE records management activity of security attributes
FMT_SMF.1	Minimal	TOE records management activity
FMT_SMR.1	Detailed	TOE records all management activities done by administrator

Table 8: Auditable Event Definitions

4.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 as specified in Part 3 of Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_FLR.1 Basic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objective
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 9: *Security Assurance Requirement* EAL2

5 TOE Summary Specification

This section summarizes the TOE specifications and provides a detailed explanation on the security functions of the TOE to meet the security functional requirements.

5.1 Cryptographic Support

The TOE supports the cryptographic services specified in the following table.

Functions	Standard
Symmetric Encryption and Decryption	
AES: 256 bits in GCM mode AES: 256 bits in CBC mode	FIPS PUB 197 NIST SP 800-140C NIST SP 800-38D
Cryptographic Key Generation	
RSA ECDSA P-256	NIST SP 800-140C
Asymmetric Encryption and Decryption	
RSA – OAEP (2048, 3072, 4096 bits)	FIPS Pub 186-4 FIPS Pub 186-5
Digital Signature Generation and Verification Services	
RSA -PKCS1 (2048, 3072, 4096 bits) ECDSA (P-256)	FIPS Pub 186-4 FIPS Pub 186-5 NIST SP 800-140C FIPS PUB 186-2
HKDF	RFC5869
Cryptographic Hashing	
SHA2-256 SHA2-384	FIPS Pub 180-4 NIST SP 800-140C FIPS PUB 202
Bcrypt	None
Message Authentication	
AES-GMAC HMAC-SHA2-256 CMAC-AES	FIPS 180-3 FIPS PUB 198-1 NIST SP 800-140C
Key Agreement Scheme ECC SSC (SP 800-56Ar3)	
ECC (P-256) 256	NIST SP 800-56A FIPS Pub 186-4 SP 800-56Ar3
KDFs	
PBKDF2	NIST SP 800-132
Cryptographic Key Destruction	
Zeroization	FIPS 140-2
Cryptographic Key Distribution	

DHE ECDHE	NIST SP 800-52r2
Scheme	
Shamir Secret Sharing	None

Table 10: Cryptographic support

The TOE implements cryptographic services listed in **Table 10** with the support of OpenSSL 1.1.1 and Intel *Software Guard Extension* (SGX). The TOE can store cryptographic keys in the CCEV database centrally. Its local vault is sealed by the Intel SGX mechanism. The system distributes the keys into LCEVs for SG-KMS's client to perform cryptographic services. The distribution process is secured by using mTLS. The TOE also provides a disposal mechanism inside an additional embedded validated module.

The cryptographic support functions below satisfy some components in the security functional requirements (SFRs) in the following manner:

- FCS_CKM.1- The TOE provides key generation services for three types of keys, namely, AES in GCM and CBC modes, RSA, and ECDSA.
- FCS_CKM.2- The TOE distributes keys between TOE and TOE users with mTLS or one-way TLS protection.
- FCS_CKM.3- The TOE provides backup for cryptographic keys using *Shamir Secret Sharing* scheme with shares distributed as stored values in physical tokens.
- FCS_CKM.4- The TOE disposes of keys that are no longer needed.
- FCS_COP.1- The TOE performs cryptographic operations as described in **Table 10: Cryptographic support** in accordance with the respective identified standards.

5.2 User Data Protection

The TOE implements authorization privilege levels for an administrator to be able to manage security attributes. The TOE also restricts access to keys that protect user data at rest by enforcing an access control.

5.2.1 Access Control

The TOE has an access control policy called **user data SFP**.

User data SFP is a policy that restricts user access. The users can be agents, administrators, and a special agent. The access restriction is based on session and slot separation mechanism. A session needs session token that is automatically obtained upon user login into the TOE's system. An agent, which is a client's application, that possesses a session token can execute cryptographic operations in the LCEV. An administrator and a special agent with a session token can execute key management services in the CCEV.

A session token consists of the following attributes: version of the session, time to live of the session, issued time of the session, session ID, client type, authorization level of

operations that should be set to low, and a distinguishable name of the application. These are packed and encrypted to form a session token. The content indicates the time limit, whether the session is versioned, and if there is another type of session token in the TOE.

The slot separation mechanism is only applicable to agents.

5.2.1.1 Access Control for an Agent

An LCEV session token is assigned to a particular agent. To get the LCEV session token, an agent must input the slot identity and its password. The application sends requests and data via TLS protocol to create a trusted path between the application and the TOE. Once the trusted path has been created, the application needs to input a session token to perform cryptographic services. A session terminates after one hour on the side of the application by locking the agent out. The session duration can be managed by an agent or an administrator. The agent can refresh the session before the TOE locks the agent's account out whereas an administrator can unlock an agent if the agent has been locked out for at least an hour. The lifetime of each session token is three (3) hours, *i.e.*, the session expires after 3 hours.

By default, an agent cannot access any slot, unless given permission by an administrator. A slot must first be assigned to a specific LCEV before an agent can be assigned access to the slot.

5.2.1.2 Access Control for an Administrator and a Special Agent

The session token of an administrator or a special agent is automatically created once they are successfully authenticated. The lifetime of the session is ten (10) minutes, without time limitation of inactivity.

The session belonging to any user can be locked if the user has failed to login three times in thirty (30) minutes. The session token is protected by a *session encryption key* (SEK).

The access control policy satisfies the following security function requirements:

- FDP_ACC.1 – The TOE provides access control protocol to prevent access from an unauthorized user.
- FDP_ACF.1 - The TOE uses security attributes in providing access control.

User data is protected by a key and the key is stored in a slot. The TOE restricts the access of user data and its attributes by using access control policies to satisfy the following security function requirements:

- FDP_DAU.1 – The TOE provides evidence to authenticate and guarantee the validity of the data.
- FDP_ETC.1 - The TOE provides access control to restrict export of external key, DEK, and user certificate without security attributes.

- FDP_ETC.2 - The TOE provides access control to restrict export of user secret with security attributes.
- FDP_ITC.1 - The TOE provides access control to restrict import of backup key without security attributes.
- FDP_ITC.2 - The TOE provides access control to restrict import of external key, DEK, user secret, user certificate with security attributes.

5.3 Identification and Authentication

Three types of users can access the TOE:

- Agents that request cryptographic services of the TOE.
- Administrators who configure and manage the TOE.
- A special agent who takes charge in case emergency operations need to be done on the TOE.

The identification and authentication security functions in the TOE enable the TOE to identify and authenticate agents, administrators, and a special agent.

5.3.1 Agent (identification and authentication)

During the configuration process, applications register their certificates to the TOE. A certificate is used as an authentication credential shortly before each communication between an application and the TOE.

The agents send requests to the TOE by using SG-KMS SDK or by calling SG-KMS RestAPI. Before the TOE fulfills the requests, the TOE implements mutual TLS which requests and verifies the application's certificate to ensure that the application has been registered. Once the agents are successfully authenticated, they can log in to the TOE.

5.3.2 Administrator (identification and authentication)

The TOE supports the identification and authentication of administrators. Each operation carried out by an administrator requires the administrator's identity and security attributes for authentication. The security attributes of an administrator are password and public certificate that is stored in a physical token, which is a smart card. Distinct operations have different privileges that require different security attributes. The operations and their respective privilege levels are defined below:

- Critical Authentication requires passwords and physical tokens from each of two administrators.

  Operations : 1. Create an administrator
  2. Reset another administrator's password
  3. Delete an administrator's account
  4. Create a new slot
  5. Delete a slot
  6. Export a DEK
  7. Revoke a DEK or user certificate

8. Delete a DEK or user certificate
9. Register an LCEV
10. Delete an LCEV
11. Renew an application's certificate
12. Verify shares
13. Regenerate a backup key
14. Register a DRC

- High Authentication requires a password and a physical token from an administrator.
  Operations : 1. Change an administrator's password
  2. Unlock another administrator account
  3. Change a slot's password
  4. Add an application to a slot
  5. Remove an application from slot
  6. Create a new DEK or user secret or user certificate
  7. Import a DEK PKCS#11 or DEK PKCS#12 or user certificate
  8. Update a DEK or user certificate
  9. Rotate a DEK
  10. Generate a CSR
  11. Add slot to an LCEV
  12. Update an LCEV
  13. Remove a slot from an LCEV
  14. Register an application
  15. Delete an application
  16. Update a user secret
  17. Delete a user secret
  18. Generate a random secret
  19. Generate a self-signed certificate
  20. Renew a self-signed certificate

- Normal Authentication requires a password from an administrator.
  Operations : 1. Show list of DEKs or user secrets or user certificates or slots or LCEVs or administrators or applications
  2. Show information of DEKs or user secrets or user certificates or slots or LCEVs or administrators or applications
  3. Check CCEV status
  4. Backup a CCEV
  5. Show the complete access control list
  6. Export a user certificate

- Low Authentication requires no attribute.

Operations: 1. Configure server
2. Get collateral

5.3.3 A Special Agent (authentication)

A special agent, collectively, holds shares of the backup key. In the event that the backup key needs to be reconstructed, enough number of shares, each retrieved from an emergency smart card, reconstruct the backup key based on the threshold *Shamir Secret Sharing*. A special agent has the privilege of performing emergency tasks. Reconstructing a backup key to restore a CCEV's database is an emergency task. Resetting the password of an administrator when all other administrators' passwords have been lost or when the quota to carry out a dual control approval cannot be met is another emergency task.

5.3.4 Password Controls

The TOE identifies two types of passwords according to their respective owners:

- Administrator's password: to authenticate an administrator.
- Slot's password: to authenticate a slot that has been created by an administrator.

Administrator's and slot's passwords must meet the following requirements:

- The length is between 8 and 64 characters.
- It contains at least one uppercase letter, one lowercase letter, one number, and one special character.
- It has no repeated adjacent characters.
- It cannot have more than two running characters of number or letter.
- It cannot have the same part as the username.

An administrator's and slot's password can be changed by another administrator by following the relevant policy.

5.3.5 Physical Token Controls (Smart Card)

The TOE identifies the following two types of smart cards:

- **Administrator smart card**
  This card serves as "what an administrator has" for authentication. The smart card contains the administrator's public certificate.
- **Emergency smart card**
  An emergency smart card contains a share of the backup key according to *Shamir Secret Sharing* scheme. The share is encrypted by using a key that is derived from the smart card's PIN.

5.3.6 Authentication Failure Handling

The TOE has the capability to configure the system to lock a user's account if the user fails to log in three times consecutively within thirty (30) minutes. There are two mechanisms to unlock the account:

- Another administrator unlocks the locked administrator account.
- The account is unlocked automatically after thirty (30) minutes.

During the hour, if the locked account has not been unlocked by another administrator, then the TOE refuses any attempts to log in from the locked administrator.

5.3.7 User Attribute Definition

Identification and authentication security functions maintain the following attributes associated with users of the TOE:

- **User identity** is the unique username identifiable by the TOE.
- **Authentication data** is the credentials used to authenticate the user's identity. They consist of
  - password,
  - certificate,
  - keypair.

The identification and authentication functions are designed to satisfy the following SFRs:

- FIA_AFL.1- The TOE is able to lock a user account after three consecutive failed attempts to log in.
- FIA_ATD.1- The TOE associates with a user identity and authentication credentials.
- FIA_SOS.1- The TOE is able to verify the password that meets password requirement.
- FIA_UAU.1 - The TOE allows several services in a CCEV to be accessed by a user before the user is authenticated.
- FIA_UAU.2 - The TOE authenticates a user in an LCEV before the user can perform any actions.
- FIA_UAU.5- The TOE implements multiple authentication mechanisms for users.
- FIA_UAU.6- The TOE is able to re-authenticate any user if deemed necessary in some cases.
- FIA_UAU.7- The TOE is able to protect any authentication feedback.
- FIA_UID.1- The TOE requires the identification of any user before the user can perform some actions in the CCEV.
- FIA_UID.2 - The TOE requires identifying the user before any action in the LCEV.

5.4 Security Management

5.4.1 Security Management Roles

The TOE supports two types of roles, namely, administrators and a special agent. Administrators and a special agent perform their respective operations through the CLI. The main difference is that a special agent only acts in emergency situations.

Each role is assigned privileges, which are termed “Access Control & Privilege” in the TOE guidance documentation. A privilege is permission to configure a TOE feature or perform an operation. Privileges are grouped into categories. The categories, roles, and attributes for each privilege are defined in Table 11: Categories, Roles, Attributes, and Privileges.

Category	Role	Authentication Attributes	Privileges
Emergency	Special agent	The TOE user is required to enter the emergency smart card PIN on the card reader keypad (out of TOE scope) to authenticate to the emergency smart card. When the emergency smart card has successfully authenticated the TOE user, the TOE shall read an SSS share in the smart card. When the TOE receives at least SSS threshold number of shares, the TOE shall be able to compute the complete secret. The computed secret shall be used as an to prove the authenticity of the TOE users. In turn, allowing the TOE users access to the corresponding privileges.	1. Restore CCEV database. 2. Reset an administrator password in case all other administrators’ passwords have been lost or when the quota to do a dual-control approval cannot be met.
Critical	Two administrators	Dual control by respective	1. Create an administrator.

		passwords and smart cards of two administrators.	2. Reset another administrator's password. 3. Delete an administrator's account. 4. Create a new slot. 5. Delete a slot. 6. Export a DEK. 7. Revoke a DEK or a user certificate. 8. Delete a DEK or a user certificate. 9. Register an LCEV. 10. Delete an LCEV. 11. Renew an application certificate. 12. Verify shares. 13. Regenerate a backup key. 14. Register a DRC.
High	An administrator	The administrator's password and smart card.	1. Change administrator's password. 2. Unlock another administrator's account. 3. Change a slot's password. 4. Add an application to a slot. 5. Remove an application from a slot. 6. Create a new DEK or secret or certificate. 7. Import a DEK PKCS#11 or DEK PKCS#12 or user certificate. 8. Update a DEK or user certificate. 9. Rotate a DEK. 10. Generate a CSR. 11. Add a slot to an LCEV. 12. Update an LCEV. 13. Remove a slot from an LCEV. 14. Register an application. 15. Delete an application. 16. Update a user secret.

			17. Delete a user secret. 18. Generate a random secret. 19. Generate a self-signed certificate. 20. Renew a self-signed certificate.
Normal	An administrator	The administrator's password.	1. Show list of DEKs or user secrets or user certificates or slots or LCEVs or administrators or applications. 2. Show information of DEKs or user secrets or user certificates or slots or LCEVs or administrators or applications. 3. Check CCEV status. 4. Backup a CCEV. 5. Show the complete access control list. 6. Export user certificate.
Low	An administrator	None	1. Configure a server. 2. Get collateral.

Table 11: Categories, Roles, Attributes, and Privileges

The TOE includes a built-in users' account. The TOE requests authentication attributes before any of the operations specified above can be performed. Actions are limited by sessions. The TOE provides only a single interface, namely the CLI, for security management.

5.4.2 Security Management Functions and Management Restrictions

The security management function provides the following capabilities for managing the behavior of the TSF, the TSF data, and their security attributes:

- Manage Administrators: An administrator with a **password** can show the list and information of all administrators.
- Manage Administrators: An administrator with a **password** and a **smart card** can change the password of and unlock another administrator's account.
- Manage Administrators: Two administrators with respective **passwords** and **smart cards** can add, delete, and reset the password of another administrator.
- Manage Administrators: A special agent with the **emergency smart cards** can reset an administrator password.

- Manage Clients' Applications: An administrator with a **password** can show the list and information about the applications.
- Manage Clients' Applications: An administrator with a **password** and a **smart card** is able to register and delete an application.
- Manage Clients' Applications: An administrator with **passwords** and **smart cards** is able to renew an application's certificate.
- Manage CCEV: An administrator with a **password** can check the status of a CCEV, backup the CCEV, and show the complete access control list.
- Manage CCEV: An administrator with a **password** and a **smart card** can request a license.
- Manage CCEV: Two administrators with respective **passwords** and **smart cards** can verify shares and create a new backup key.
- Manage CCEV: Two administrators with respective **passwords** and **smart cards** can register a DRC site.
- Manage LCEVs: An administrator with a **password** can show the list and information of the LCEVs.
- Manage LCEVs: An administrator with a **password** and a **smart card** can add a slot to an LCEV, remove a slot from an LCEV, and update an LCEV.
- Manage LCEVs: Two administrators with respective **passwords** and **smart cards** are able to register and delete an LCEV.
- Manage slots: An administrator with a **password** can show the list and information regarding the slots. The information includes each slot id, slot label, and creation date.
- Manage slots: An administrator with a **password** and **a smart card** can change a slot's password, add an application to a slot, and remove an application from a slot.
- Manage slots: Two administrators with respective **passwords** and **smart cards** are able to add and delete a slot.
- Manage DEKs: An administrator with a **password** can show the list and information of keys.
- Manage DEKs: An administrator with a **password** and a **smart card** can add a DEK, import a DEK, update a DEK, rotate a DEK, and generate a CSR.
- Manage DEKs: Two administrators with respective **passwords** and **smart cards** can export, revoke, and delete a DEK.
- Manage secrets: An administrator with **a password** can show the list and information of the secrets.
- Manage secrets: An administrator with **a password** and a **smart card** can add, update, and delete a secret.
- Manage secrets: An administrator with **a password** and a **smart card** can generate a random secret.
- Manage certificates: An administrator with **a password** can show the list and information about the certificates and export a certificate.
- Manage certificates: An administrator with **a password** and a **smart card** can generate a self-signed certificate, import a certificate, and renew a self-signed certificate.

- Manage certificates: Two administrators with respective **passwords** and **smart cards** can revoke and delete a certificate.

The security management function is designed to satisfy the following security functional requirements:

- FMT_MSA.1 – The TOE enforces user data SFP to restrict the management of security attributes.
- FMT_SMF.1 – The TOE provides the capability to manage the user and TSF data.
- FMT_SMR.1 – The TOE defines roles based on the authentication attributes and privileges that they have.
- FMT_MTD.1 – The TOE restricts access to TSF data to administrators.

5.5 TSF Protection

To ensure consistent interpretation of security attributes between the TOE and business applications, TSF data is interpreted in the following manner:

- REST API for security attributes of external key and user secret
- Binary for security attributes of AES DEK
- PEM for security attributes of RSA DEK and ECDSA DEK
- PEM for user certificates.

The TSF protection function is designed to satisfy the following security functional requirements:

- FPT_TDC.1 – The TOE consistently interprets the security attributes of user data which are transferred between the TOE and its user.

5.6 TOE Access

The TOE implements some session-based configuration termination:

- **Agent's session**
  The TOE sets an `auto logout` mechanism for the application. The mechanism is determined by a session in the LCEV. The TOE assigns a session for the application, when it logs in to the TOE, in the form of a session token. The lifetime of the session is three (3) hours, and the session needs to be refreshed manually by the agent before an hour. After the lifetime, the session stops being valid, which means that the agent must log in again. This session is encrypted by using the *session encryption key* (SEK), which is rotated every four (4) hours. The SEK is generated inside each LCEV and the LCEV stores three SEKs. An SEK becomes usable five (5) minutes after it has been generated. To support this mechanism, the server should support the NTP. An NTP server stops working when its battery is low, causing an out-of-sync time. The SEK generation mechanism ensures the security of the TOE should this failure occurs.

- **Administrator's session**

The TOE set an `auto logout` mechanism for the administrators. The mechanism is determined by a session in the CLI. The TOE assigns a session for an administrator, when she or he logs in to the TOE, in the form of a session token. The lifetime of the session is ten (10) minutes before an automatic logout. Administrators use session tokens that they obtain after logging in to their respective accounts to create a session. Each session token is encrypted by a *session encryption key* (SEK). The SEK is generated in the CCEV and rotates every four (4) hours.

All sessions are set by the TOE. The protocol is used to terminate the interactive session. The TOE access function is to satisfy the following *security functional requirements*:

- FTA_SSL.3 – The TOE terminates administrator and agent session.

5.7 Trusted Path

The TOE supports a trusted path for clients' applications to communicate with the TOE. The trusted path is implemented using TLS 1.3 for access to the LCEV. An agent initiates the trusted path by sending a request by using SG-KMS RestAPI or SG-KMS SDK.

The trusted path is used for initial authentication and all subsequent administrative actions. The trusted path is designed to satisfy the following *security functional requirement*:

- FTP_TRP.1 – The TOE provides a trusted path for agents, administrators, and a special agent to communicate with the TOE.
- FCS_COP.1(10) - The TOE provides SHA-256 and SHA-382 hashing algorithms to support mTLS and one-way TLS.
- FCS_COP.1(11) - The TOE provides AES-128-GCM and AES-256-GCM hashing algorithms to support mTLS and one-way TLS.
- FCS_CKM.1(4) - The TOE implements HKDF for key derivation as part of mTLS and one-way TLS.
- FCS_CKM.2(1) and FCS_CKM.2(2) - The TOE implements DHE and ECDHE for key distribution as part of mTLS and one-way TLS.

5.8 Resource Utilization

The TOE consists of two logically separated main components, namely, the CCEV and the LCEV. The CCEV performs key management functions such as key generation, distribution, rotation, revocation, recovery, expiry, export, import, and disposal. The LCEV performs cryptographic services such as encrypt, decrypt, seal, unseal, sign, verify, tokenize, detokenize, generate MAC, and verify MAC. The objects that are involved in cryptographic services are replicated from the CCEV to the LCEV.

Cryptographic services in LCEV are designated to run normally with or without the CCEV. In turn, cryptographic services shall be maintained even when the CCEV is down. The drawback is that, when the CCEV is down, no user can add or manage the keys. The LCEV remains capable in performing cryptographic services using existing keys.

The resource utilization function is designed to satisfy the following security functional requirements:

- FRU_FLT.1 – The TOE can ensure cryptographic operations even when the CCEV processor breaks down.

5.9 Security Audit

The TOE records every administrative activity in the CCEV. For each event, the record contains the date, time, and type, the identity of each relevant administrator, and the outcome, complete with the error codes. An HMAC is then generated on each event's record to guarantee its integrity. The output log file can be transferred from the Audit Log server into the CCEV, allowing any administrator to export or archive the record as needed.

The security audit function is designed to satisfy the following security functional requirement:

- FAU_GEN.1 – TOE can generate audit records.

6 Rationale

This section explains the rationale for completeness and consistency of the *Security Target*. The rationale includes the following area:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification

6.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. The following table shows the detail:

RATIONALE	O.I_A	O.I_A_HANDLING	O.KEY_ACC	O.PASS_CONTROLS	O.PROTECT_COMMS	O.SECURE_MANAGE	O.SESS_TERM	O.CRYPTO	O.DATA_AU	O.DEGRADED_OP	O.AUDIT	OE.COMPET_USERS	OE.PHYSICAL_ENV	OE.IT_SUPPORT_COMP	OE.RELIABLE_TIME
T.DATA_COMP	X		X		X	X	X								
T.UNAUTH_ACC		X		X											
A.COMPET_USERS												X			
A.PHYSICAL_ENV													X		
A.SECURE_PLATFORM														X	
A.SMART_CARD														X	
A.SC_READER														X	
A.RELIABLE_TIME															X
A.AUDIT_SERVER														X	
OSP.CRYPTO								X							
OSP.DATA_AU									X						
OSP.DEGRADED_OP										X					
OSP.AUDIT											X				X

Table 12: Security Problem Definition to Security Objective Correspondence

6.1.1 T.DATA_COMP

***An unauthorized user may compromise the confidentiality and integrity of user and/or TSF data being communicated between TOE users and TOE*.** This threat is countered by the following security objectives:

- O.I_A address this threat by requiring all TOE users to be identified and authenticated before accessing user and/or TSF data or security management functions. This verifies the identity and authenticity of the TOE users interacting with the TOE. Hence, reducing the risk of unauthorized users compromising the confidentiality and integrity of the user data and/or TSF data.
- O.KEY_ACC addresses this threat by providing access control to the DEK, user secret and user certificate. This controls TOE user access to user data and/or TSF data according to a set of access control rules. This further reduces the exposure of user data and/or TSF data.
- O.PROTECT_COMMS addresses this threat by protecting the integrity and confidentiality of user and/or TSF data being communicated between TOE users and TOE. This reduces the exposure of user data and/or TSF data in transit.
- O.SECURE_MANAGE addresses this threat by enforcing access control to its security management functions. This reduces the risk of exposure on TSF data.
- O.SESS_TERM addresses this threat by terminating the user session. This reduces the risk of exposure on user data and/or TSF data.

6.1.2 T.UNAUTH_ACC

An unauthorized user tries to access the TSF data through repeated password-guessing attempts. This threat is countered by the following security objectives:

- O.I_A_HANDLING addresses this threat by increasing the difficulty of applying brute-force attack.
- O.PASS_CONTROLS addresses this threat by improving the security strength of users' password. In turn, reducing the likelihood of a successful brute-force attack.

6.1.3 A.COMPET_USERS

TOE users are trusted and competent. This assumption is upheld by the following security objective:

- OE.COMPET_USERS directly upholds this assumption.

6.1.4 A.PHYSICAL_ENV

The TOE is deployed in a physically secured environment. This assumption is countered by the following security objective:

- OE.PHYSICAL_ENV directly upholds this assumption.

6.1.5 A.SECURE_PLATFORM

The underlying platform is trusted and secure. This assumption is upheld by the following security objective:

- OE.IT_SUPPORT_COMP directly upholds this assumption.

6.1.6 A.SMART_CARD

The user smart card is trusted and secure. This assumption is upheld by the following security objective:

- OE.IT_SUPPORT_COMP directly upholds this assumption.

6.1.7 A.SC_READER

The smart card reader is trusted and secure. This assumption is upheld by the following security objective:

- OE.IT_SUPPORT_COMP directly upholds this assumption.

6.1.8 A.RELIABLE_TIME

A time server shall be deployed to provide reliable timestamp to the TOE. This assumption is upheld by the following security objective:

- OE.RELIABLE_TIME directly upholds this assumption.

6.1.9 A.AUDIT_SERVER

An audit server shall be deployed to store the audit logs generated by the TOE. The server is trusted and secure. This assumption is upheld by the following security objective:

- OE.IT_SUPPORT_COMP directly upholds this assumption.

6.1.10 OSP.CRYPTO

The TOE shall implement the cryptographic algorithms stated in Table 10. This OSP is upheld by the following security objective:

- O.CRYPTO directly upholds this OSP.

6.1.11 OSP.DATA_AU

The TOE shall generate evidence that can be used as a guarantee of the validity of user data at rest. This OSP is upheld by the following security objective:

- O.DATA_AU directly upholds this OSP.

6.1.12 OSP.DEGRADED_OP

The TOE shall ensure that LCEV remains operational even when CCEV is non-operational. This OSP is upheld by the following security objective:

- O. DEGRADED_OP directly upholds this OSP.

6.1.13 OSP.AUDIT

The TOE shall log security-relevant events. This OSP is upheld by the following security objective:

- O.AUDIT directly upholds this OSP.

6.2 Security Functional Requirements Rationale

This section identified all security functional requirements in this *Security Target* and shows the correlation with objectives. More detail is shown in **Table 13**.

SFR/ Security objectives	O.I_A	O.I_A_HANDLING	O.KEY_ACC	O.PASS_CONTROLS	O.PROTECT_COMMS	O.SECURE_MANAGE	O.SESS_TERM	O.CRYPTO	O.DATA_AU	O.DEGRADED_OP	O.AUDIT
FCS_COP.1(1)								**X**			
FCS_COP.1(14)								**X**			
FCS_COP.1(2)								**X**			
FCS_COP.1(3)								**X**			
FCS_COP.1(4)								**X**			
FCS_COP.1(5)								**X**			
FCS_COP.1(6)								**X**			
FCS_COP.1(7)								**X**			
FCS_COP.1(8)								**X**			
FCS_CKM.1(1)								**X**			
FCS_CKM.1(2)								**X**			
FCS_CKM.3								**X**			
FCS_CKM.4								**X**			
FCS_CKM.1(3)								**X**			
FCS_COP.1(9)								**X**			
FCS_COP.1(13)								**X**			

FCS_COP.1(10)					**X**			**X**			
FCS_COP.1(11)					**X**			**X**			
FCS_COP.1(12)								**X**			
FCS_CKM.1(4)					**X**			**X**			
FCS_CKM.2(1)					**X**			**X**			
FCS_CKM.2(2)					**X**			**X**			
FCS_CKM.4								**X**			
FDP_ACC.1			**X**								
FDP_ACF.1			**X**								
FDP_DAU.1									**X**		
FDP_ETC.1			**X**								
FDP_ETC.2			**X**								
FDP_ITC.1			**X**								
FDP_ITC.2			**X**								
FDP_RIP.1					**X**						
FIA_AFL.1		**X**									
FIA_ATD.1	**X**										
FIA_SOS.1				**X**							
FIA_UAU.1	**X**										
FIA_UAU.2	**X**										
FIA_UAU.5	**X**										
FIA_UAU.6	**X**										

FIA_UAU.7	**X**										
FIA_UID.1	**X**										
FIA_UID.2	**X**										
FMT_MSA.1						**X**					
FMT_SMF.1						**X**					
FMT_SMR.1						**X**					
FMT_MTD.1						**X**					
FRU_FLT.1										**X**	
FTA_SSL.3(1)					**X**		**X**				
FTA_SSL.3(2)					**X**		**X**				
FTP_TRP.1					**X**						
FPT_TDC.1			**X**								
FRU_FLT.1										**X**	
FAU_GEN.1											**X**

Table 13: Objective to Requirement Correspondence

6.2.1 O.I_A

The TOE shall require all users of the TOE to be identified and authenticated before gaining access to user and/or TSF data or security management functions. The following security functional requirements contribute to satisfying this security objective:

FIA_ATD.1(1)	specify the identification authentication security attributes that are required to be maintained by the TOE to enforce FIA_UID.1, FIA_UAU.1 and FIA_UAU.2.
FIA_ATD.1(2)	
FIA_ATD.1(3)	
FIA_UAU.1	specifies authentication of administrator and special agent.
FIA_UAU.2	specifies authentication of agent.
FIA_UID.1	specifies identification of administrator and special agent

FIA_UID.2	specifies identification of agent
FIA_UAU.5	specifies multiple authentication mechanisms for operations with varying privilege levels; operations with higher privileged levels would require more robust authentication mechanisms.
FIA_UAU.6	specifies re-authentication when TOE user requests for higher privileged operations than the currently granted.
FIA_UAU.7	specifies feedback during authentication, in turn, reducing possible exposure of authentication data.

6.2.2 O.I_A_HANDLING

The TOE shall limit the number of failed identification and authentication attempts. The following security functional requirements contribute to satisfying this security objective:

FIA_AFL.1	directly upholds the security objective.

6.2.3 O.KEY_ACC

The TOE shall provide access control to manage DEKs, user secrets, and user certificates to authorized users only. The following security functional requirements contribute to satisfying this security objective:

FDP_ACC.1	specifies the access control policy and its corresponding rules that is applied to agents, FDP_ACCs, operations, and user data.
FDP_ACF.1	
FDP_ITC.1	FDP_ACC.1 and FDP_ACF.1 are applied for import of user data.
FDP_ITC.2	FDP_ACC.1 and FDP_ACF.1 are applied for import of user data.
FDP_ETC.1	FDP_ACC.1 and FDP_ACF.1 are applied for export of user data.
FDP_ETC.2	FDP_ACC.1 and FDP_ACF.1 are applied for export of user data.
FPT_TDC.1	ensures consistent interpretation of security attributes during import of user data.

6.2.4 O.PASS_CONTROLS

The TOE shall check users against a quality metric to improve password strength. The following security functional requirements contribute to satisfying these security objectives:

FIA_SOS.1	directly upholds the security objective.

6.2.5 O.PROTECT_COMMS

The TOE shall protect the confidentiality and integrity of user and/or TSF data being communicated between the TOE and TOE user. The following security functional requirements contribute to satisfying these security objectives:

FCS_COP.1(10)	specifies the cryptographic algorithm for hashing operations
FCS_COP.1(11)	specifies the cryptographic algorithm for symmetric encryption/decryption operations
FCS_CKM.2(1)	specify the key distribution methods for supporting FTP_TRP.1
FCS_CKM.2(2)	
FTP_TRP.1	specifies the trusted path between TOE and TOE users.
FTA_SSL.3(1)	specifies the termination of interactive session after a stipulated duration for trusted paths between TOE and TOE users. This reduces the exposure of non-active sessions.
FTA_SSL.3(2)	
FDP_RIP.1	specifies the destruction of transient user certificates and user secret after use. This reduces the exposure of user certificates and user secret.

6.2.6 O.SECURE_MANAGE

The TOE shall enforce access control to its security management functions. The following security functional requirements contribute to satisfying these security objectives:

FMT_MSA.1	restricts the management of security attributes to administrators and special agents.
FMT_SMF.1	defines the security management functions
FMT_SMR.1	maintains the security roles of administrator, special agent and agent.

FMT_MTD.1	restricts the management of TSF data to an administrator

6.2.7 O.SESS_TERM

The TOE shall provide a mechanism to terminate a user session. The following security functional requirements contribute to satisfying these security objectives:

FTA_SSL.3(1)	directly upholds the security objective.
FTA_SSL.3(2)	

6.2.8 O.CRYPTO

The TOE shall implement the cryptographic algorithms stated in Table 10. The following security functional requirements contribute to satisfying these security objectives:

FCS_COP.1(1)	specify the cryptographic algorithms stated in **Table 10**.
FCS_COP.1(14)	
FCS_COP.1(2)	
FCS_COP.1(3)	
FCS_COP.1(4)	
FCS_COP.1(5)	
FCS_COP.1(6)	
FCS_COP.1(7)	
FCS_COP.1(8)	
FCS_CKM.1(1)	
FCS_CKM.1(2)	
FCS_CKM.3	
FCS_CKM.1(9)	
FCS_COP.1(13)	
FCS_COP.1(10)	
FCS_COP.1(11)	
FCS_COP.1(12)	
FCS_CKM.1(4)	
FCS_CKM.2(1)	
FCS_CKM.2(2)	

FCS_CKM.4	specifies the destruction of cryptographic keys after use. This reduces the exposure of cryptographic keys.

6.2.9 O.DATA_AU

The TOE shall generate evidence that can be used as a guarantee of the validity of user data at rest. The following security functional requirements contribute to satisfying these security objectives:

FDP_DAU.1	directly upholds the security objective.

6.2.10 O.DEGRADED_OP

The TOE shall ensure that LCEV remains operational even when CCEV is non-operational. The following security functional requirements contribute to satisfying these security objectives:

FRU_FLT.1	directly upholds the security objective.

6.2.11 O.AUDIT

The TOE shall log security-relevant events. The following security functional requirement contributes to satisfying this security objective:

FAU_GEN.1	directly upholds the security objective.

6.3 Security Assurance Requirements Rationale

EAL 2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. The TOE is targeted at an environment with good physical access security where it is assumed that attackers have BASIC attack potential. Therefore, the target assurance of EAL 2 is appropriate for such an environment.

6.4 Requirement Dependency Rationale

The following table identifies the SFRs claimed in the ST, their dependencies as defined in CC Part 2, and how the dependency is satisfied in the ST. The following table shows the dependencies:

SFR	Dependencies	Fulfilment
FCS_COP.1(1)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. The TOE relies on the underlying Intel SGX to generate DEK.
	FCS_CKM.4	FCS_CKM.4

FCS_COP.1(14)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. The TOE relies on the underlying Intel SGX to generate DEK.
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(2)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(1)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(3)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(4)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(1)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(5)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. Cryptographic hashing does not require key.
	FCS_CKM.4	Not applicable. Cryptographic hashing does not require key.
FCS_COP.1(6)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. The TOE relies on the underlying Intel SGX to generate the DEK
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(7)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. The TOE relies on the underlying Intel SGX to generate DEK.
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(8)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. The TOE relies on the underlying Intel SGX to generate DEK.
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(1)	FCS_CKM.2 / FCS_COP.1	FCS_COP.1(2), FCS_COP.1(4)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(2)	FCS_CKM.2 / FCS_COP.1	FCS_COP.1(3)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM.1(4), FDP_ITC.1
FCS_CKM.3	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FDP_ITC.1
	FCS_CKM.4	FCS_CKM.4

FCS_CKM.1(3)	FCS_CKM.2 / FCS_COP.1	FCS_COP.1(9)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(9)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(3)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(13)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	Not applicable. Cryptographic hashing does not require key.
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(10)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(4)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(11)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(4)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(12)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(4)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(4)	FCS_CKM.2 / FCS_COP.1	FCS_COP.1(11), FCS_COP.1(12)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2(1)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(4)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2(2)	FCS_CKM.1/ FDP_ITC.1 / FDP_ITC.2	FCS_CKM.1(4)
	FCS_CKM.4	FCS_CKM.4
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1
	FMT_MSA.1	FMT_MSA.1
FDP_DAU.1	None	Not appliable
FDP_ETC.1	FDP_ACC.1/ FDP_IFC.1	FDP_ACC.1
FDP_ETC.2	FDP_ACC.1/ FDP_IFC.1	FDP_ACC.1
FDP_ITC.1	FDP_ACC.1/ FDP_IFC.1	FDP_ACC.1
	FMT_MSA.3	Not applicable. The TOE does not implement this function. All security attributes of objects must be configured to a defined value on creation.

FDP_ITC.2	FDP_ACC.1/ FDP_IFC.1	FDP_ACC.1
	FTP_ITC.1/FTP_TRP.1	FTP_TRP.1
	FPT_TDC.1	FPT_TDC.1
FDP_RIP.1	None	Not applicable.
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	None	Not applicable.
FIA_SOS.1	None	Not applicable.
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.2	FIA_UID.1	FIA_UID.2 is hierarchical to FIA_UID.1
FIA_UAU.5	None	Not applicable.
FIA_UAU.6	None	Not applicable.
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.1	None	Not applicable.
FIA_UID.2	None	Not applicable.
FMT_MSA.1	FDP_ACC.1/FDP_IFC.1	FDP_ACC.1
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_SMF.1	None	Not applicable.
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FRU_FLT.1	FPT_FLS.1	Not applicable. LCEV operates independently of CCEV; LCEV operations shall maintain even when CCEV non-operational.
FTA_SSL.3	None	Not applicable.
FTP_TRP.1	None	Not applicable.
FPT_TDC.1	None	Not applicable.
FAU_GEN.1	FPT_STM.1	FPT_STM.1 is upheld by OE.RELIABLE_TIME.

Table 14: Requirement Dependencies

6.5 TOE Summary Specification Rationale

This section describes the TOE Summary Specification, in detail describing how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function specification.

RATIONALE	Cryptographic Support	User Data Protection	Identification & Authentication	Security Management	TSF Protection	TOE Access	Trusted Path	Resource Utilization	
FCS_CKM.1(*)	X								
FCS_CKM.2(*)	X								
FCS_CKM.3	X								
FCS_CKM.4	X								
FCS_COP.1(*)	X								
FDP_ACC.1		X							
FDP_ACF.1		X							
FDP_DAU.1		X							
FDP_ETC.1		X							
FDP_ETC.2		X							
FDP_ITC.1		X							
FDP_ITC.2		X							
FDP_RIP.1		X							
FIA_AFL.1			X						
FIA_ATD.1			X						
FIA_SOS.1			X						

FIA_UAU.1(*)			**X**						
FIA_UAU.2			**X**						
FIA_UAU.5			**X**						
FIA_UAU.6			**X**						
FIA_UAU.7			**X**						
FIA_UID.1			**X**						
FIA_UID.2			**X**						
FMT_MSA.1				**X**					
FMT_SMF.1				**X**					
FMT_SMR.1				**X**					
FMT_MTD.1				**X**					
FTA_SSL.3 (*)						**X**			
FTP_TRP.1							**X**		
FPT_TDC.1					**X**				
FRU_FLT.1								**X**	
FAU_GEN.1									**X**

Table 15: TOE Summary Specification Rationale

7 Appendix

In this evaluation, the TOE provides deployment in two architecture models. In the first model, key management services and cryptographic services take place in one physical machine. The second model separates the two services in distinct physical machines. Configuration between parts of TOE allows the TOE to continue to operate in a fully secure manner in the event of a failure of a TOE's part.

User data and TSF data are guaranteed confidentiality by using AES, RSA, and ECDSA. To detect modification, the TOE uses HMAC-SHA-256, HMAC-SHA384, HMAC-SHA512, AES-GMAC, and key recalculation. The specific choice depends on the data. The flow of data between parts of the TOE or between the TOE and users take place over TLS and remote attestation to ensure confidentiality and integrity of the transmitted data. If the TOE detects data modification, then the TOE closes the connection down.

TSF data is generated in the CCEV, but it is used in the LCEV to perform cryptographic services. TSF data and its attributes are replicated and streamed from the CCEV to the LCEV. The replication data flow over mTLS and remote attestation before being saved into a trusted area in the LCEV (SGX enclave). If the replicating or flowing of the data fails, it automatically rolls the replication back. Replication is automated during heartbeat if there are up to twenty five different versions of the database in the CCEV and the LCEV. If there are more than eight different versions, then the CCEV and the LCEV are declared "out of sync" and an administrator must do the replication manually.
The following data resides in the database that is replicated:

- Slots and their attributes
- Keys and their attributes
- Secrets and their attributes
- Certificates and their attributes
- Applications' information and its attributes

The CCEV is designated as a central time server, with the overall system relying on the CCEV for timing. The LCEV also communicates regularly with the CCEV through a heartbeat checking mechanism. The TOE maintains time using secure NTP to synchronize them. This mechanism is also used as a timestamp record in each audit record. The time configuration is handled by the TOE.

7.1 Intel SGX

The TOE utilizes Intel SGX, which lies outside the scope of this evaluation, to strengthen the protection to the user data at rest, in transit, and in use. Intel SGX is an extra set of Intel CPU instructions. The TOE calls Intel SGX to create **SGX enclaves**. These are areas in a running program that are protected by hardware to enforce confidentiality and integrity, even when privileged operating systems have been compromised. An SGX enclave constitutes a trusted environment for data at rest.

The TOE performs **remote attestation** to protect data in transit. The attestation guarantees that a TOE component is indeed communicating with an enclave which is hosted in a CPU with Intel

SGX capability and that the enclave runs a specific piece of software. In the TOE, remote attestation is implemented during the initialization stage in each communication between CCEV and LCEV and between CCEV and the Audit log service. The attestation depends on PPID (*Platform Provisioning Identity*) to authenticate either the hardware that hosts a TOE component itself or the hardware it communicates with.

Should the need arise to store user data in an untrusted media, the TOE deploys the SGX **sealing mechanism** to encrypt the data prior to secure storage.

7.2 Key Structure

The TOE implements a user data protection security function to control user access to cryptographic keys. The TOE uses cryptographic keys to protect user plaintexts, DEKs, user secrets, and user certificates. The keys come in several types, depending on **Figure 3: Key Structure**:

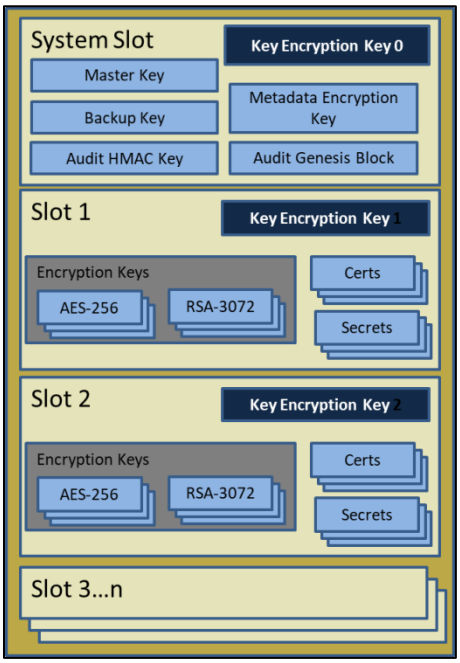

Figure 3: Key Structure

- System keys: the keys never leave the TOE and are only for internal usage of SG-KMS. These system keys consist of:
  - *Metadata encryption key* (MEK) that protects the metadata of an encryption process.
  - *Key encryption key* (KEK) that protects a specific slot.
  - The master key that provides the top-level protection for all keys.
  - The backup key that protects the master key during backup.

  The keys are generated during provisioning.
- *Data encryption keys* (DEK): these keys serve specific purposes, *e.g.*, data encryption, key wrapping, and digital signing. The keys are generated by using AES-GCM, RSA, and ECDSA P-256 algorithms. They are protected by a *key encryption key* (KEK) that never leaves the TOE.

- Externally stored keys: the keys leave the TOE and are stored in the database of the client's applications. The keys are protected by using a wrapping key that never leaves the TOE.
- Smart card keys: the keys that are generated in the TOE for storage in the TOE and the relevant smart cards. These keys are used to protect data in-transit between the TOE and the smart cards. The smart card keys come in two types:
    - Secure channel keys that protect both the data and a key which are transferred between the TOE and the smart cards. A secure channel key contains:
        - an encryption key,
        - a MAC key,
        - a smart card DEK.
    - Smart card channel session keys that protect the secure channel keys.

We note that an AES DEK whose purpose is encryption can be rotated and exported. This DEK is periodically rotated to derive a new version of such a DEK. The rotation can be set into automatic or manually by an administrator.