Utimaco
Enterprise Secure Key Manager
Security Target

Version 1.0
19 February, 2019

Prepared for:
Utimaco

1160 Enterprise Way
Sunnyvale CA, 94089

Prepared By:
Leidos
Accredited Testing & Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

Table of Contents

LIST OF TABLES

1. Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is Enterprise Secure Key Manager (ESKM), version 5.1, from Utimaco. The ESKM provides capabilities for generating, storing, serving, controlling and auditing access to data encryption keys. It enables organizations to protect and preserve access to business-critical, sensitive, data-at-rest encryption keys, both locally and remotely.

The Security Target contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Utimaco Enterprise Secure Key Manager Security Target

ST Version – 1.0

ST Date – 19 February, 2019

TOE Identification – Enterprise Secure Key Manager, version 5.1

TOE Developer – Utimaco

Evaluation Sponsor – Utimaco

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

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

This ST and the TOE it describes are conformant to the following package:

- EAL2 Augmented (ALC_FLR.2).

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements—Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FMT_MTD.1(1) and FMT_MTD.1(2) indicate that the ST includes two iterations of the FMT_MTD.1 requirement.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST—Other sections of the ST use bolding and/or different fonts (such as `Courier` and Arial) to highlight text of special interest, such as captions, commands, or attributes specific to the TOE.

1.4 Glossary

This ST uses a number of terms that have a specific meaning within the context of the ST and the TOE. This glossary provides a list of those terms and how they are to be understood within this ST.

Administrator	In the context of this ST, an administrator is defined as a user of the TOE that has a local administrator account on the TOE and can login to the TOE via the CLI or Management Console. The administrator's capabilities on the TOE are determined by the privileges (or "access controls") assigned to the administrator. An administrator with all privileges is termed a High Access Administrator.
Client	In the context of this ST, a client is defined as a user of the TOE that connects to the TOE using either the ESKM XML protocol or the KMIP in order to request key management services. Clients typically are encrypting client devices (e.g., storage systems) or applications (e.g., databases).
ESKM XML protocol	XML-based protocol used by clients to communicate to the Key Management Server (KMS) component of the TOE to request operations on ESKM keys.
KMIP	Key Management Interoperability Protocol—a communication protocol that defines message formats for the manipulation of cryptographic keys on a key management server.

1.5 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document. A brief definition is provided for abbreviations that are potentially unfamiliar, are specific to the TOE, or not obviously self-explanatory.

AES	Advanced Encryption Standard
CA	Certificate Authority
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration Management
CRL	Certificate Revocation List
DES	Data Encryption Standard
DNS	Domain Name System

ESKM	Enterprise Secure Key Manager
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IT	Information Technology
KMS	Key Management Service
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
MAC	Message Authentication Code
MIB	Management Information Base—a database used for managing the entities in a communications network; usually associated with SNMP
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
RSA	Rivest-Shamir-Adleman—an asymmetric cryptographic algorithm
SAR	Security Assurance Requirement
SCP	Secure copy
SFP	Security Function Policy
SFR	Security Functional Requirement
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
UID	Unit Identifier
XML	Extensible Markup Language

2. TOE Description

The TOE is Enterprise Secure Key Manager (ESKM), version 5.1, from Utimaco. The TOE provides capabilities for generating, storing, serving, controlling and auditing access to data encryption keys. It enables organizations to protect and preserve access to data-at-rest encryption keys, both locally and remotely.

The remainder of this section provides an overview of the TOE and a description of the TOE, including a description of the physical and logical scope of the TOE.

2.1 TOE Overview

The TOE is an appliance that provides security policy and key management services to encrypting client devices and applications. After enrollment, clients (such as storage systems, application servers and databases) make requests to the TOE for creation and management of cryptographic keys and related metadata.

In its evaluated configuration, the TOE comprises two or more ESKM appliances configured as a single cluster, which provides redundancy and allows the TOE to continue to operate in a fully secure fashion in the event of a failure of a node in the cluster. Clustering also enables multiple ESKMs in a distributed environment to synchronize and replicate configuration information, which reduces administration overhead. Nodes in a cluster communicate with each other to maintain a synchronized configuration. Communications between nodes in a cluster occur over TLS 1.2.

The TOE supports two methods for servicing client requests—Key Management Service (KMS) and Key Management Interoperability Protocol (KMIP). Each method implements its own access control policy that determines who can perform operations on the objects within the scope of the policy—keys for KMS and managed objects for KMIP. KMIP managed objects include keys, certificates, and user-defined objects. Both the KMS and KMIP methods support TLS for client communications.

Administrators can configure and manage the TOE remotely via a web-based Graphical User Interface (GUI) or a Command Line Interface (CLI). The administrator uses HTTPS to access the GUI and Secure Shell (SSH) to connect to the CLI. The TOE also has a serial console port, but this is intended for use only during initial installation and configuration of the TOE. Administrators require privileges (also termed “access controls” in the TOE documentation) in order to configure a TOE feature or perform an operation. The TOE defines High Access Administrators, which are administrators with all privileges assigned (the built-in “admin” user is a High Access Administrator). A High Access Administrator can create other administrators and assign privileges to them.

All TOE users (administrators and clients) must be successfully identified and authenticated by the TOE before gaining access to any other TOE services. The TOE supports password and certificate-based authentication mechanisms. The TOE provides capabilities to configure minimum strength requirements (e.g., minimum length, required character sets) for passwords. The TOE can be configured to track the number of consecutive failed authentication attempts and block further authentication attempts for a configurable time period when the configured threshold has been met. The TOE will terminate interactive sessions that have been idle for a configurable period of time.

The TOE is able to generate audit records of security-relevant events occurring on the TOE, including startup and shutdown of the TOE, successful and unsuccessful administrator login attempts, and key management activities. It provides administrators with the ability to review audit records stored in the audit trail. The audit records are stored on the TOE appliance, where they are protected from unauthorized modification and deletion.

2.2 TOE Architecture

2.2.1 Deployment Architecture

The TOE is intended to be deployed in an enterprise’s network infrastructure where it can provide key generation and management services to encrypting client devices and applications. As described above, the evaluated configuration comprises two or more appliances in a cluster configuration. The appliances in a cluster do not need to be collocated—the only requirement is they are able to communicate with each other over TCP/IP (communications between nodes in fact are protected using TLS).

In addition to interfaces for client key management services and clustering, the TOE supports Administration, Monitoring, and IT Services interfaces.

The Administration interface consists of the CLI (accessible remotely via SSH) and GUI (accessible via HTTPS).

The Monitoring interface supports the following capabilities:

- KMS Health Check—allows clients to check the availability of the KMS service by sending the TOE an HTTP request on a configured IP address and port
- KMIP Health Check—similarly allows clients to check the availability of the KMIP service by sending the TOE an HTTP request on a configured IP address and port
- FIPS Status—allows clients to check the TOE's FIPS status (FIPS mode enabled or not enabled) and the most recent results of the FIPS self-tests
- Syslog—the TOE supports the ability to export audit and log records to an external syslog server for off-line storage and review
- SNMP Agent—the TOE can be configured to provide SNMP data (MIBs, traps) to an external network management station (NMS)
- SIEM—the TOE can be configured to export audit and log records to an external SIEM solution.

The IT Services interface supports the following capabilities:

- NTP—the TOE can be configured to synchronize its time by polling an external NTP server
- Backup—the TOE can be configured to export log files and backup files to an external server using SCP.

The TOE can also be configured to use an LDAP server to support remote authentication, but this capability is not included in the scope of the evaluation.

Figure 1 depicts a typical deployment of the TOE. In this example, the TOE is represented as a two-node cluster. Note that the nodes can be collocated or installed at physically separate locations.

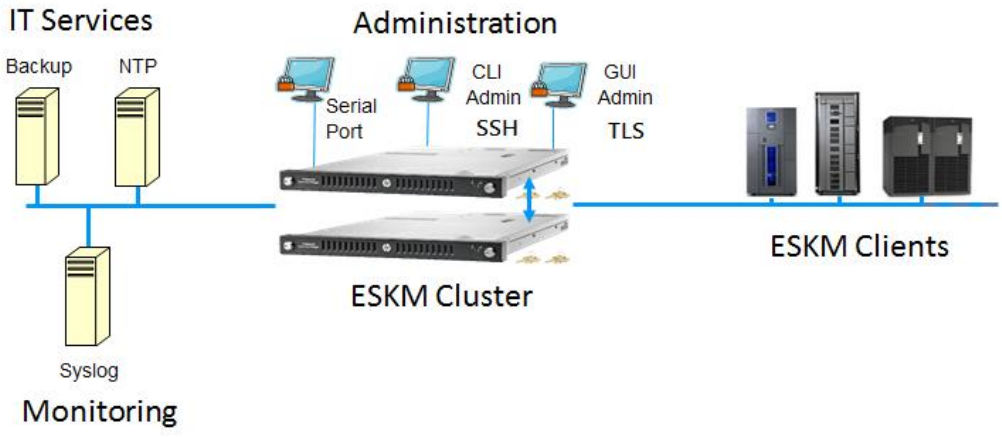

Figure 1: Example TOE Deployment

2.2.2 Software Architecture

The TOE software comprises the following major components:

- KMS Server—provides key management services to ESKM clients that access the TOE using the ESKM XML protocol
- KMIP Server—provides key management services to KMIP clients that access the TOE using KMIP
- Authentication Services—support the KMS and KMIP servers by handling requests for client authentication

- Certificate Services—provides a local CA and certificate services for the TOE
- Restricted Shell—implements the administrator CLI
- GUI Webserver—implements the Management Console
- Cluster Server—manages communication and synchronization with other nodes in the cluster
- Health Webserver—provides a health check service that enables external entities to query the status of the KMS and KMIP servers and the FIPS status of the TOE
- Log/Event services—handles communications with external monitoring services such as syslog and SNMP
- Core—comprises the underlying operating system, file system, database system, cryptographic services and core services made available to the other software components of the TOE.

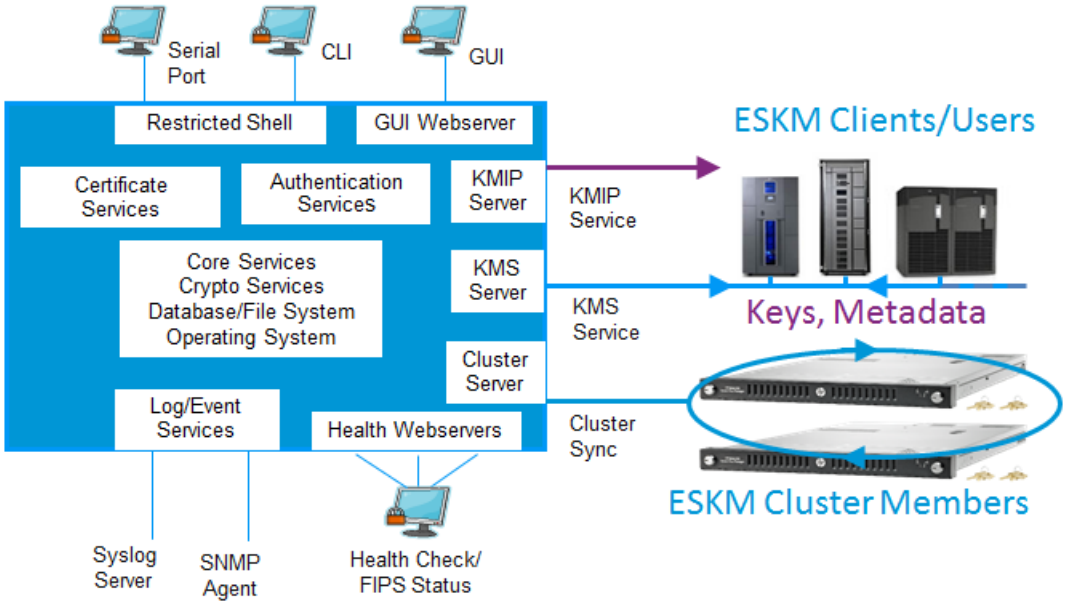

Figure 2: TOE Software Architecture

2.2.3 Physical Boundaries

The physical boundary of the TOE is the ESKM appliance. It is a single rack-mountable chassis that takes up 1 rack unit of space. It provides the following external interfaces:

- 1 RS-232C DB9 serial port
- 1 VGA DB15 monitor port
- 2 autosensing 10/100/1000 BASE-T (Ethernet) RJ-45 ports
- Status LEDs:
    - Unit Identifier (UID) (front)
    - Power/Standby
    - Aggregate Network
    - System Health
    - NIC activity
    - NIC link
    - UID (rear).

Use of the TOE may require the following components in its operational environment:

- Serial terminal client, connected via the serial console port, to support initial configuration of the TOE via the CLI
- Management client PCs, connected via a network port, to support remote management of the TOE. The management client PC in turn requires:
  - A browser to connect to the Management Console; and/or
  - An SSHv2 client to connect to the CLI
- Syslog server for remote storage of audit records
- NTP server to provide system clock synchronization
- SCP server for remote storage of backups and log files.

Note, when configured, the syslog server receives audit records as they are generated by the TOE. The TOE can also be configured to periodically transfer its log files to a remote host via SCP.

2.2.4 Logical Boundaries

This section summarizes the security functions provided by the TOE.

2.2.4.1 Security Audit

The TOE is able to generate audit records of security-relevant events occurring on the TOE, including startup and shutdown of the TOE, successful and unsuccessful administrator login attempts, and key management activities. It provides administrators with the ability to review audit records stored in the audit trail. The audit records are stored on the TOE appliance, where they are protected from unauthorized modification and deletion.

2.2.4.2 Cryptographic Support

The TOE provides the following key management services to external clients: key generation (symmetric key and asymmetric key pairs); key distribution; key storage; and key destruction. The TOE uses cryptographic protocols to protect communications: between nodes in a cluster (TLS); with external IT entities (TLS); and with remote administrators (SSH access to CLI, HTTPS access to GUI). In support of these protocols, the TOE can perform the following cryptographic operations: symmetric encryption and decryption using AES; digital signature generation and verification using RSA; cryptographic hashing using SHA-1; and keyed-hash message authentication using HMAC.

2.2.4.3 User Data Protection

The TOE implements an access control policy on KMS keys and a separate access control policy on KMIP objects. Access to KMS keys is based on ownership and group membership. Access to KMIP objects is based on user group membership and permissions to operate on members of object groups.

2.2.4.4 Identification and Authentication

The users of the TOE comprise ***administrators***, who manage the TOE and its configuration, and ***clients***, who request key management services from the TOE. Clients are classified as ESKM clients or KMIP clients, depending on the protocol used to access the TOE—ESKM XML for ESKM clients and KMIP for KMIP clients.

The TOE identifies and authenticates all users of the TOE before granting them access to the TOE. The TOE associates a user identity and authentication data (password and/or certificate) with each client and user identity, password and privileges (or “access controls”) with each administrator. The TOE enforces minimum requirements for the construction of user passwords and provides a mechanism to lock a user account after a configured number of consecutive failed attempts to logon.

2.2.4.5 Security Management

The TOE implements a privilege-based security management model. Administrators are granted privileges to perform security management functions on the TOE. Each privilege grants access to a specific subset of the security

management capabilities of the TOE. An administrator with all privileges is termed a High Access Administrator and is able to perform all security management functions, including creating and managing other administrator accounts and changing user and administrator passwords.

2.2.4.6 TSF Protection

In its evaluated configuration, the TOE comprises two or more ESKM appliances configured as a single cluster, which provides redundancy and allows the TOE to continue to operate in a fully secure fashion in the event of a failure of a node in the cluster. Communications between nodes in a cluster occur over TLS, which provides confidentiality and detection of modification of transmitted data.

The TOE includes its own time source for providing reliable time stamps that are used in audit records.

2.2.4.7 TOE Access

The TOE will terminate interactive sessions after a period of inactivity configurable by an administrator.

The TOE allows administrators to terminate their own interactive sessions.

2.2.4.8 Trusted Channel/Path

The TOE provides a trusted path for TOE administrators to communicate with the TOE. The trusted path is implemented using HTTPS for access to the GUI and SSHv2 for access to the CLI. Administrators initiate the trusted path by establishing an HTTPS connection (using a supported web browser) or SSH session (using an SSH client). The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS or SSHv2 ensures all communication over the trusted path is protected from disclosure and undetected modification.

2.2.5 Excluded Functionality

When operating in FIPS mode, the ESKM cannot use non-FIPS-approved algorithms for cryptographic operations. TLS connections using non-approved TLS/SSL ciphersuites are not supported and only FIPS-approved keys can be created. Cryptographic algorithms other than those identified in section 6.1.8.1 are not supported in the evaluated configuration. This will also exclude KMIP's ECDSA key pair generation functionality and the TOE's ECDHE key agreement support. The ESKM will also disable global keys, FTP, LDAP, and SSL 3.0.

The TOE's configurable ability to limits SSH administrative login attempts is outside the scope of the evaluation.

2.3 TOE Documentation

This section identifies the guidance documentation included in the TOE.

- *Utimaco Enterprise Secure Key Manager v5.1.0 Software v7.1.0 User's Guide*, October 2018, Part Number M6H81-9002E
- *Utimaco Enterprise Secure Key Manager 5.1 Installation and Replacement Guide*, October 2018, Part Number M6H81-9001B
- *Utimaco Enterprise Secure Key Manager v5.1.0 - Software Version 7.1.0 Release Notes*, October 2018
- *HP Enterprise Secure Key Manager Key Protection Best Practices*, 4AA2-1403ENW, rev. 4, March 2011.

3. Security Problem Definition

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

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

A.PROTECT	The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

3.2 Threats

This section identifies and describes the threats to be countered by the TOE and its operational environment.

T.BRUTE_FORCE	An unauthorized user may gain access to the TOE through repeated password-guessing attempts.
T.DATA_COMPROMISE	Data on long-term storage media may be compromised if control of that media passes to unauthorized entities.
T.KEY_COMPROMISE	Encrypted data may be compromised if unauthorized users gain access to encryption keys.
T.NETWORK_COMPROMISE	TSF data communicated between components of the TOE, or between the TOE and external entities, is disclosed or undetectably modified.
T.NO_ACCOUNTABILITY	Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.
T.UNATTENDED_SESSION	An unauthorized user gains access to the TOE via an unattended authorized user session.
T.UNAUTHORIZED_ACCESS	Unauthorized users gain access to the TOE and its services.
T.UNAUTHORIZED_ACTIVITY	Authorized users perform unauthorized actions on the TOE.
T.UNAVAILABILITY	Authorized users are unable to access TOE services due to failure of the TOE.
T.UNRECOVERABLE_DATA	Encrypted data may be unrecoverable if encryption keys are mishandled.

4. Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

O.AUDIT	The TOE shall be able to generate audit records of security-relevant events, identifying users causing the events as applicable.
O.AUDIT_REVIEW	The TOE shall provide a means for authorized users to review the audit records generated by the TOE.
O.AUDIT_STORAGE	The TOE shall protect stored audit records from unauthorized modification and deletion.
O.CRYPTOGRAPHY	The TOE shall perform cryptographic operations to support protocols used to protect data in transit.
O.HIGH_AVAILABILITY	The TOE shall provide the capability to continue servicing user requests after a failure of part of the TOE.
O.I_&_A	The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.
O.KEY_ACCESS	The TOE shall restrict access to managed encryption keys to authorized users.
O.KEY_MANAGEMENT	The TOE shall provide services for authorized users to request generation of encryption keys and shall provide management services for those keys.
O.PASSWORD_CONTROLS	The TOE shall provide a mechanism to reduce the likelihood that users choose weak passwords.
O.PROTECTED _COMMS	The TOE shall protect communications between distributed parts of the TOE, and between the TOE and external entities, from disclosure and undetected modification.
O.SECURITY_MANAGEMENT	The TOE shall restrict the ability to perform security management functions on the TOE to authorized administrators having appropriate privileges.
O.SESSION_TERMINATION	The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.
O.THROTTLE	The TOE shall limit the rate at which consecutive unsuccessful authentication attempts can be performed.

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE:

OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
OE.PERSONNEL	Those responsible for the TOE must ensure that personnel working as authorized administrators have been carefully selected and trained for proper operation of the TOE.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. SFRs were drawn exclusively from Part 2 of the Common Criteria v3.1 Revision 4.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_STG.1: Protected audit trail storage
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
FIA: Identification and Authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.2: User authentication before any action
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_UID.2: User identification before any action
FMT: Security Management	FMT_MOF.1: Management of security functions behavior
	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialisation
	FMT_MTD.1: Management of TSF data
	FMT_REV.1: Revocation
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TOE Security Functions	FPT_FLS.1: Failure with preservation of secure state
	FPT_ITT.1: Basic internal TSF data transfer protection
	FPT_ITT.3: TSF data integrity monitoring
	FPT_STM.1: Reliable time stamps
FTA: TOE Access	FTA_SSL.3: TSF-initiated termination
	FTA_SSL.4: User-initiated termination

Requirement Class	Requirement Component
FTP: Trusted path/channels	FTP_ITC.1: Trusted channel
	FTP_TRP.1: Trusted path

Table 1: TOE Security Functional Components

5.1.1 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 [***not specified***] level of audit; and
c) [**The following auditable events:**
- **Administrator logins and logouts**
- **All modifications of TSF data**
- **All client requests, including login and logout**
- **Service starts, stops, and restarts**
- **Cluster replication operations**
- **Changes to the system clock**].

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**].

FAU_GEN.2 – User identity association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_SAR.1 – Audit review

FAU_SAR.1.1 The TSF shall provide [**administrators**] with the capability to read [**all auditable events that are recorded**] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_STG.1 – Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [***prevent***] unauthorised modifications to the stored audit records in the audit trail.

5.1.2 Cryptographic Support (FCS)

FCS_CKM.1 – Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**DRBG (AES in CTR mode)**] and specified cryptographic key sizes [**128, 192, 256 bits for AES; 2048 bits for RSA**] that meet the following: [**SP 800-90A**].

FCS_CKM.2 – Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [**ESKM XML protocol; KMIP**] that meets the following: [**none for ESKM XML protocol; "Key Management Interoperability Protocol Specification Version 1.4", OASIS Standard, 22 November 2017 for KMIP**].

FCS_CKM.3 – Cryptographic key access

FCS_CKM.3.1 The TSF shall perform [**key storage**] in accordance with a specified cryptographic key access method [**ESKM XML protocol; KMIP**] that meets the following: [**none for ESKM XML protocol; "Key Management Interoperability Protocol Specification Version 1.4", OASIS Standard, 22 November 2017 for KMIP**].

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 [**key zeroization**] that meets the following: [**FIPS 140-2**].

FCS_COP.1(1) – Cryptographic operation (symmetric encryption and decryption)

FCS_COP.1.1(1) The TSF shall perform [**symmetric encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES**] and cryptographic key sizes [**128, 192, 256 bits in ECB, CBC, CTR, GCM and KW modes; 256 bits in CCM mode**] that meet the following: [**FIPS 197**].

FCS_COP.1(2) – Cryptographic operation (digital signature generation and verification)

FCS_COP.1.1(2) The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**2048, 3072 bits**] that meet the following: [**FIPS 186-4**].

FCS_COP.1(3) – Cryptographic operation (cryptographic hashing)

FCS_COP.1.1(3) The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512**] and cryptographic ~~key~~ **hash** sizes [**160, 224, 256, 384 and 512 bits**] that meet the following: [**FIPS 180-3**].

FCS_COP.1(4) – Cryptographic operation (keyed-hash message authentication)

FCS_COP.1.1(4) The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512**] and cryptographic key sizes [**Key Size < Block Size, Key Size > Block Size**] that meet the following: [**FIPS 198-1, FIPS 180-3**].

5.1.3 User Data Protection (FDP)

FDP_ACC.1(1) – Subset access control (ESKM SFP)

FDP_ACC.1.1(1) The TSF shall enforce the [**ESKM SFP**] on [

- **Subjects: ESKM Users; Administrators**
- **Objects: ESKM Keys**
- **Operations: Create; Import; Export; Query; Delete**].

FDP_ACC.1(2) – Subset access control (KMIP SFP)

FDP_ACC.1.1(2) The TSF shall enforce the [**KMIP SFP**] on [

- **Subjects: KMIP Users**
- **Objects: KMIP Objects**
- **Operations: Supported KMIP Operations**].

FDP_ACF.1(1) – Security attribute based access control (ESKM SFP)

FDP_ACF.1.1(1) The TSF shall enforce the [**ESKM SFP**] to objects based on the following: [

- **Subject security attributes:**
  - **ESKM Users – Identity**
  - **Administrators – Privilege**
- **Object security attributes:**
  - **ESKM Keys – Owner, Deletable, Exportable, Groups**].

FDP_ACF.1.2(1) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- **An ESKM User can Create an ESKM Key and will be the ESKM Key Owner**
- **An ESKM User that is not the Owner can Export an ESKM Key if the ESKM User is a member of a Group that has Export permission for the ESKM Key and the ESKM Key has the Exportable attribute**
- **An ESKM User that is not the Owner can Delete an ESKM Key if the ESKM User is a member of a Group that has Delete permission for the ESKM Key and the ESKM Key has the Deletable attribute**].

FDP_ACF.1.3(1) The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [

- **High Access Administrators and Administrators with the 'Keys and Authorization Policies' privilege can perform all operations on all ESKM Keys**
- **The ESKM Key Owner can Export the ESKM Key if it has the Exportable attribute**
- **The ESKM Key Owner can Delete the ESKM Key if it has the Deletable attribute**].

FDP_ACF.1.4(1) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**no explicit deny rules**].

FDP_ACF.1(2) – Security attribute based access control (KMIP SFP)

FDP_ACF.1.1(2) The TSF shall enforce the [**KMIP SFP**] to objects based on the following: [

- **Subject security attributes:**
  - **KMIP Users – Identity, User Groups**
- **Object security attributes:**
  - **KMIP Objects – Identity, Object Group**].

FDP_ACF.1.2(2) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- **Each KMIP User is a member of at least one User Group**
- **User Groups list the Object Groups the KMIP Users in the User Group can access, and the permissions the KMIP Users have for each listed Object Group**
- **A KMIP User can perform a Supported KMIP Operation on a KMIP Object if and only if:**
  - **The KMIP User is a member of a User Group that lists an Object Group containing the KMIP Object, AND**
  - **The User Group grants permission to perform the requested Supported KMIP Operation on an Object Group containing the KMIP Object**].

FDP_ACF.1.3(2) The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**no explicit authorize rules**].

FDP_ACF.1.4(2) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**no explicit deny rules**].

5.1.4 Identification and Authentication (FIA)

FIA_AFL.1 – Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [***an administrator configurable positive integer within [1 .. 5]***] unsuccessful authentication attempts occur related to [**ESKM User and KMIP User login**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [***met***], the TSF shall [**block further authentication attempts for an administrator-configured number of seconds**].

FIA_ATD.1 – User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [

- **User identity**
- **Authentication data**
- **Privileges]**.

FIA_SOS.1 – Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [

a) the following minimum requirements:

- **passwords must contain at least 5 different characters**
- **passwords must not contain only whitespace**
- **passwords must not resemble a phone number, dictionary word, or reversed dictionary word**
- **passwords must not be based on the username associated with the password**
- **passwords must contain at least one non-alphanumeric character**

b) the following requirements when configured by an administrator

- **passwords must be a minimum length**
- **passwords must contain at least one lower case alphabetic character**
- **passwords must contain at least one upper case alphabetic character**
- **passwords must contain at least one numeric character**
- **passwords must contain at least one special character**].

FIA_UAU.2 – User authentication before any action

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.

FIA_UAU.5 – Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [**password mechanism, certificate mechanism**] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [**following rules:**

- **Administrators are authenticated using the password mechanism**
- **ESKM Users are authenticated using the password mechanism. If configured by an administrator, an ESKM User is additionally authenticated using the certificate mechanism**
- **KMIP Users are authenticated using the certificate mechanism. If configured by an administrator, a KMIP User is additionally authenticated using the password mechanism**
- **Any user configured for multiple authentication mechanisms must satisfy the authentication requirements of both mechanisms in order to be successfully authenticated**].

FIA_UID.2 – User identification before any action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.1.5 Security Management (FMT)

FMT_MOF.1(1) – Management of security functions behavior (Audit)	
FMT_MOF.1.1(1)	The TSF shall restrict the ability to [***determine the behavior of, modify the behaviour of***] the functions [**audit**] to [**High Access Administrator, Administrator with 'Logging' privilege**].

FMT_MOF.1(2) – Management of security functions behavior (Backup)	
FMT_MOF.1.1(2)	The TSF shall restrict the ability to [***determine the behavior of, modify the behaviour of***] the functions [**backup**] to [**High Access Administrator, Administrator with 'Backup' privileges**].

FMT_MOF.1(3) – Management of security functions behavior (Restore)	
FMT_MOF.1.1(3)	The TSF shall restrict the ability to [***determine the behavior of, modify the behaviour of***] the functions [**restore**] to [**High Access Administrator, Administrator with 'Restore' privileges**].

FMT_MOF.1(4) – Management of security functions behavior (TLS)	
FMT_MOF.1.1(4)	The TSF shall restrict the ability to [***determine the behavior of, modify the behaviour of***] the functions [**TLS**] to [**High Access Administrator, Administrator with 'SSL' privilege**].

FMT_MOF.1(5) – Management of security functions behavior (Connection timeout, account locking)	
FMT_MOF.1.1(5)	The TSF shall restrict the ability to [***determine the behavior of, enable, disable***] the functions [**Connection timeout, User Account Lockout**] to [**High Access Administrator, Administrator with 'KMS/KMIP Server' privilege**].

FMT_MSA.1(1) – Management of security attributes (ESKM SFP)	
FMT_MSA.1.1(1)	The TSF shall enforce the [**ESKM SFP**] to restrict the ability to [***query, modify***] the security attributes [**all ESKM Key attributes**] to [**High Access Administrator, Administrator with 'Keys and Authorization Policies' privilege**].

FMT_MSA.1(2) – Management of security attributes (KMIP SFP)	
FMT_MSA.1.1(2)	The TSF shall enforce the [**KMIP SFP**] to restrict the ability to [***change_default, query, modify, delete, [create]***] the security attributes [**User Groups, Object Groups**] to [**High Access Administrator, Administrator with 'Users and Groups' privilege**].

FMT_MSA.3(1) – Static attribute initialization (ESKM SFP)	
FMT_MSA.3.1(1)	The TSF shall enforce the [**ESKM SFP**] to provide [***permissive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2(1)	The TSF shall allow the [**High Access Administrator, Administrator with 'Keys and Authorization Policies' privilege, ESKM Key Owner**] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3(2) – Static attribute initialization (KMIP SFP)	
FMT_MSA.3.1(2)	The TSF shall enforce the [**KMIP SFP**] to provide [***permissive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2(2)	The TSF shall allow the [**KMIP User creating the KMIP object**] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1(1) – Management of TSF Data (User accounts)

FMT_MTD.1.1(1) The TSF shall restrict the ability to [***modify, delete, [create]***] the [**ESKM Users, KMIP Users**] to [**High Access Administrator, Administrator with 'Users and Groups' privilege**].

FMT_MTD.1(2) – Management of TSF Data (Administrators)

FMT_MTD.1.1(2) The TSF shall restrict the ability to [***modify, delete, [create]***] the [**Administrators**] to [**High Access Administrator**].

FMT_MTD.1(3) – Management of TSF Data (Certificates)

FMT_MTD.1.1(3) The TSF shall restrict the ability to [***delete, [create]***] the [**Certificates**] to [**High Access Administrator, Administrator with 'Certificates' privilege**].

FMT_MTD.1(4) – Management of TSF Data (Certificate Authorities)

FMT_MTD.1.1(4) The TSF shall restrict the ability to [***modify, delete, [create]***] the [**Certificate Authorities**] to [**High Access Administrator, Administrator with 'Certificate Authorities' privilege**].

FMT_MTD.1(5) – Management of TSF data (Cluster configuration)

FMT_MTD.1.1(5) The TSF shall restrict the ability to [***modify, delete, [create]***] the [**cluster configuration**] to [**High Access Administrator, Administrator with 'Cluster' privilege**].

FMT_MTD.1(6) – Management of TSF data (Passwords, password controls)

FMT_MTD.1.1(6) The TSF shall restrict the ability to [***modify***] the [**password controls, other users' passwords**] to [**High Access Administrator**].

FMT_MTD.1(7) – Management of TSF data (NTP, date/time settings)

FMT_MTD.1.1(7) The TSF shall restrict the ability to [***modify***] the [**NTP settings, system date/time**] to [**High Access Administrator, Administrator with 'NTP and Date/Time' privilege**].

FMT_REV.1 – Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke [**administrator privileges**] associated with the [***users***] under the control of the TSF to [**High Access Administrator**].

FMT_REV.1.2 The TSF shall enforce the rules [**immediately**].

FMT_SMF.1 – Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [

- **Manage ESKM and KMIP Users**
- **Manage Administrators**
- **Manage ESKM SFP security attributes**
- **Manage KMIP SFP security attributes**
- **Manage certificates**
- **Manage Certificate Authorities**
- **Manage connection timeout**
- **Manage password controls**
- **Manage audit security function**
- **Manage cluster configuration**
- **Change passwords**
- **Manage NTP and date/time settings**
- **Manage User Account Lockout**
- **Configure TLS**
- **Backup and restore configurations**].

FMT_SMR.1 – Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [

- **High Access Administrator**
- **Administrator with one or more of the following privileges:**
  - **Keys and Authorization Policies**
  - **Users and Groups**
  - **Certificates**
  - **Certificate Authorities**
  - **SSL**
  - **KMS/KMIP Server**
  - **Cluster**
  - **Network and Date/Time**
  - **Logging**
  - **Backup privileges**
  - **Restore privileges**].

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

5.1.6 Protection of the TOE Security Functions (FPT)

FPT_FLS.1 – Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [**failure of a node in the cluster**].

FPT_ITT.1 – Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from [***disclosure***] when it is transmitted between separate parts of the TOE.

FPT_ITT.3 – TSF data integrity monitoring

FPT_ITT.3.1 The TSF shall be able to detect [***modification of data***] for TSF data transmitted between separate parts of the TOE.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions: [**close connection with the other part of the TOE**].

FPT_STM.1 – Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.1.7 TOE Access (FTA)

FTA_SSL.3 – TSF-initiated termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [**period of inactivity, depending on the session type, as follows:**

- **For ESKM Users—between 1 and 7200 seconds (default 3600)**
- **For KMIP Users—between 1 and 7200 seconds (default 3600)**
- **For CLI—between 1 and 720 minutes (default 30)**
- **For Management Console—60 minutes**].

FTA_SSL.4 – User-initiated termination

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

5.1.8 Trusted Path/Channels (FTP)

FTP_ITC.1 – Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**copying TOE backup configuration to trusted IT product**].

FTP_TRP.1 – Trusted path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [***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 [***disclosure, [undetected modification]***].

FTP_TRP.1.2 The TSF shall permit [***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, [all remote administrative actions]***].

5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.2 components as specified in Part 3 of the 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 a CM system
	ALC_CMS.2: Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.2: Flaw reporting procedures
ASE: Security Target evaluation	ASE_CCL.1: Conformance claims
	ASE_ECD.1: Extended components definition
	ASE_INT.1: ST introduction
	ASE_OBJ.2: Security objectives
	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

Requirement Class	Requirement Component
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

Table 2: EAL 2 augmented with ALC_FLR.2 Assurance Components

5.2.1 Development (ADV)

ADV_ARC.1 – Security architecture description	
ADV_ARC.1.1D	The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2D	The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
ADV_ARC.1.3D	The developer shall provide a security architecture description of the TSF.
ADV_ARC.1.1C	The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2C	The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3C	The security architecture description shall describe how the TSF initialisation process is secure.
ADV_ARC.1.4C	The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5C	The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2 – Security-enforcing functional specification	
ADV_FSP.2.1D	The developer shall provide a functional specification.
ADV_FSP.2.2D	The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.2.1C	The functional specification shall completely represent the TSF.
ADV_FSP.2.2C	The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.2.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.2.4C	For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.2.5C	For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.
ADV_FSP.2.6C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.
ADV_TDS.1 – Basic design	
ADV_TDS.1.1D	The developer shall provide the design of the TOE.
ADV_TDS.1.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.1.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C	The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C	The design shall describe the behavior of each SFR-supporting or SFR non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.
ADV_TDS.1.4C	The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.
ADV_TDS.1.5C	The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C	The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.
ADV_TDS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.2.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance

AGD_OPE.1.1D	The developer shall provide operational user guidance.
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C	The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6C	The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle Support (ALC)

ALC_CMC.2 – Use of a CM system	
ALC_CMC.2.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2D	The developer shall provide the CM documentation.
ALC_CMC.2.3D	The developer shall use a CM system.
ALC_CMC.2.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage	
ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery procedures	
ALC_DEL.1.1D	The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.2 – Flaw reporting procedures	
ALC_FLR.2.1D	The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2D	The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3D	The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.2.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.2.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.2.5C	The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C	The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.
ALC_FLR.2.7C	The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.2.8C	The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_FLR.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims

ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.
ASE_CCL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended components definition

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 – ST introduction

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarise the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide a security objectives rationale.
ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.
ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D	The developer shall provide a statement of security requirements.
ASE_REQ.2.2D	The developer shall provide a security requirements rationale.
ASE_REQ.2.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition	
ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.
ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification	
ASE_TSS.1.1D	The developer shall provide a TOE summary specification.
ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.1.2E	The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.2.5 Tests (ATE)

ATE_COV.1 – Evidence of coverage	
ATE_COV.1.1D	The developer shall provide evidence of the test coverage.
ATE_COV.1.1C	The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing	
ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.
ATE_FUN.1.1C	The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C	The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C	The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent testing – sample

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E	The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

AVA_VAN.2.1D	The developer shall provide the TOE for testing.
AVA_VAN.2.1C	The TOE shall be suitable for testing.
AVA_VAN.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.2.2E	The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.2.3E	The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
AVA_VAN.2.4E	The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions implemented by the TOE to satisfy the SFRs.

6.1 TOE Security Functions

The TOE implements the following security functions that together satisfy the SFRs claimed in Section 5.1 of this ST:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- TSF Protection
- TOE Access
- Trusted Channel/Path.

6.1.1 Security Audit

6.1.1.1 Log Files and Auditable Events

The TOE maintains the following files that record security-relevant and other system events, including administrative actions, network activity, and cryptography requests:

- System—records the following auditable events:
  - TOE startup and shutdown
  - Service starts, stops, and restarts
  - Cluster replication operations
  - Changes to the system clock resulting from NTP synchronization
- Audit—records the following auditable events:
  - Administrator login and logout
  - All configuration changes (modification to TSF data)
- Activity—records each request received by the KMS server, including client login and logout and key export
- KMIP—records each request received by the KMIP server, including client login and logout and key export
- Client Event—records all client requests containing the <RecordEventRequest> element.

Each audit record generated by the TOE includes the following information: date and time of the event; the event type; the outcome of the event; and the responsible subject (including the user identity where applicable).

An administrator with **Logging** privilege can schedule log rotations, configure the number of logs archived on the TOE, stipulate the maximum log file size, and transfer logs to a log server.

6.1.1.2 Audit Review

The TOE provides all administrators with the ability to review the contents of the audit trail, both at the Management Console and via the CLI.

When an administrator successfully logs on via the Management Console, the Home Page is viewed. The Home Page includes a display of the most recent entries in the Audit log and also provides a link to the Audit Log Viewer page. Alternatively, the administrator can select the Device tab in the Management Console and then choose the Log Viewer page for each of the logs maintained by the TOE. Each Log Viewer page displays the most recent entries in the relevant log file (the number displayed is configurable, with a default of 10).

An administrator logged in to the CLI can use the `show` command to display the contents of a specific log.

6.1.1.3 Log Storage and Rotation

For each type of log, the current log entries are kept in a file named **Current**. The TOE rotates log files based on a configured rotation schedule or file size. When a log file is rotated, the **Current** log file is closed and renamed with a timestamp. This renamed file is then either stored in the log archive or transferred off the TOE, depending on the configuration. A new **Current** log file is then created.

An administrator with **Logging** privilege can configure the rotation schedule to automatically rotate logs on a daily, weekly, or monthly basis, at any time of day. The TOE maintains these settings for each log type. For example, the Activity and Audit logs can be configured to rotate on different schedules. When a log is rotated, the TOE also attempts to transfer the log to a transfer destination, if one has been configured for it. Only the SCP protocol is supported for log transfer in the evaluated configuration.

In addition, the administrator can specify a maximum log file size, which causes logs to be rotated when they reach a certain size, regardless of their rotation schedule. For example, the administrator can schedule the TOE to rotate the Audit Log every Sunday morning at 3:15 or when the file size reaches 100 MB, whichever comes first.

The log files are stored in the TOE's file system. They are not accessible to client users and are accessible to administrators only via the commands provided by the Management Console and CLI. All administrators are authorized to delete System, Activity, KMIP and Client Event logs. The TOE does not provide a mechanism to delete Audit logs.

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

- FAU_GEN.1—the TOE generates audit records for all specified auditable events, including starting and stopping the TOE (equivalent to starting up and shutting down the audit functions). Furthermore, each audit record includes the date and time of the event, the event type, the outcome of the event, and the responsible subject.
- FAU_GEN.2—for auditable events resulting from the actions of identified users, the TOE includes the user identification as the responsible subject in the generated audit record.
- FAU_SAR.1—the TOE provides all administrators with the capability to read the contents of each of the log files comprising the audit trail. The contents of log files are human-readable and can be viewed using either the Management Console GUI or the CLI.
- FAU_STG.1—the TOE protects the log files comprising the audit trail from unauthorized modification and deletion.

6.1.2 Cryptographic Support

The TOE is a FIPS 140-2 validated cryptomodule (Certificate #2862). It provides the cryptographic services specified in the following table.

Functions	Standards	Certificates
Symmetric Encryption and Decryption		
AES: 128, 192, 256 bits in ECB, CBC, CTR, GCM and KW modes; 256 bits in CCM mode	FIPS PUB 197 NIST SP 800-38A NIST SP 800-38D	AES #5951
Triple DES: 168 bits (3-key) in ECB and CBC modes	FIPS 46-3	Triple-DES #2899
Digital Signature Generation and Verification Services		
RSA (2048, 3072 bits)	FIPS PUB 186-4	RSA #C 235
Cryptographic Hashing		

Functions	Standards	Certificates
SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	FIPS Pub 180-4	SHS #4703
Keyed-Hash Message Authentication Code		
HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512: (Key Size < Block Size, Key Size > Block Size)	FIPS Pub 198-1 FIPS Pub 180-3	HMAC #3923
Deterministic Random Bit Generation		
DRBG (AES in CTR mode)	NIST SP 800-90A	DRBG #2501, 2502

Table 3: Cryptographic Services

The TOE implements the cryptographic services listed in **Table 3** above both for its own uses (e.g., in support of TLS and SSH protocols) and to satisfy requests for cryptographic services from KMIP clients. The KMIP server also supports CMAC mode authentication for 3DES and AES operations; and ECDSA-256 and ECDSA-384 X.509 certificates to support Suite B Cryptography however these features have not been subject to evaluation. In addition to these services, the TOE is able to store cryptographic keys and to distribute them to ESKM clients and KMIP clients using the ESKM XML protocol and KMIP respectively. When keys are to be destroyed (e.g., when no longer required, or at request of an ESKM client or KMIP client), they are overwritten with zeroes in accordance with the requirements of FIPS 140-2.

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

- FCS_CKM.1—the TOE uses its implementation of DRBG using AES in CTR mode to generate symmetric Triple DES and AES keys and RSA key pairs.
- FCS_CKM.2—the TOE distributes keys to ESKM clients and KMIP clients using the ESKM XML protocol and KMIP respectively.
- FCS_CKM.3—the TOE provides secure key storage services to ESKM and KMIP clients.
- FCS_CKM.4—the TOE destroys keys that are no longer required by overwriting with zeroes.
- FCS_COP.1(*)—the TOE performs cryptographic operations as specified in **Table 3** above, using the identified algorithms in accordance with the identified standards.

6.1.3 User Data Protection

The TOE implements the User Data Protection security function to control access to cryptographic keys.

The TOE supports two types of cryptographic keys:

- ESKM keys—these are keys created and managed using KMS. They also can be managed by administrators with appropriate privileges via the Management Console
- KMIP objects—these are keys and other cryptographic objects created and managed using the KMIP protocol.

The TOE provides mechanisms to convert between ESKM keys and KMIP objects.

The TOE implements separate access control policies for ESKM keys and KMIP objects.

6.1.3.1 ESKM Access Control

ESKM keys are keys that are created and managed using KMS. In addition, an Administrator with the **Keys and Authorization Policies** privilege can create and manage ESKM keys via the Management Console.

An ESKM key is composed of two main parts: the key bytes and the key metadata. The key bytes are the bytes used by the cryptographic algorithm to encrypt or decrypt data. The key metadata contains information about the key bytes:

key name; owner username; algorithm; key size; creation date; group permissions; and custom attributes. The metadata also indicates if the key is a versioned key, deletable, or exportable.

ESKM keys can be global or owned by a particular user. Global keys are keys that are available to everyone, with no authentication required. Global keys are not supported in the evaluated configuration.

An authenticated ESKM user submits a request via the ESKM XML protocol to the TOE to create an ESKM key. The user requesting ESKM key generation becomes the owner of the ESKM key. The key generation request can also specify if the key is to be exportable or deletable. An exportable key is a key that an authorized client can export from the TOE. A deletable key is a key that an authorized client can delete from the TOE.

The owner of an ESKM key has full access to the key. Other users can be granted access to the key via membership of an ESKM user group. An administrator with **Keys and Authorization Policies** privilege can assign ESKM user groups to an ESKM key, and then set each group's permissions for the key. The TOE defines two ESKM group permissions:

- Export—the Export permission is applicable only if the key is exportable, and takes one of three values:
    - Always—members of the group can always perform an export operation on the key
    - Never—members of the group cannot export the key
    - Authorization policy—members of the group can always perform an export operation on the key according to the terms of the specified authorization policy
- Full—Full permission allows members of the group to perform the same key operations available to the key owner. Therefore, key export is only allowed if the key is exportable. Similarly, key deletion is only allowed if the key is deletable. The Full permission takes one of two values:
    - Always—members of the group can always perform key operations available for the key
    - Never—members of the group cannot perform any operations on the key (this allows the administrator to remove a previously set Full permission for a group without having to delete the group from the key's Group Permissions table).

Note that setting Full permission to Always overrides any authorization policy set for the Export permission. That is, if the key is exportable and the group has Full permission, then a group member will always be able to export the key, regardless of the settings in the authorization policy. Also, a user that is a member of multiple groups assumes the union of the group permissions.

6.1.3.2 KMIP Access Control

KMIP objects are keys and other cryptographic objects created and managed using KMIP. KMIP users first authenticate to the KMIP Server component of the TOE and then submit requests for operations on KMIP objects. The KMIP Server determines if the requested operation is permitted, based on the access control rules of the KMIP access control policy.

The KMIP access control policy is based on group memberships. There are two types of group defined within the scope of the KMIP access control policy—User Groups and Object Groups.

- A KMIP User Group contains only KMIP Users. All KMIP Users must belong to at least one KMIP User Group. User Group membership is configured when an administrator creates the KMIP User. The administrator can change user group membership from either the Management Console or the CLI.
- A KMIP Object Group contains only KMIP Objects.

Since the KMIP access control model is group-based, all KMIP Objects in the same Object Group can be accessed by all users that have permission to access that group. Each KMIP User is configured with a default Object Group. All KMIP Objects created by a KMIP User are placed in the user's default Object Group, unless the user specifies a different group in the create request. Administrators cannot transfer KMIP Objects from one Object Group to another. This can only be done by KMIP Users with the appropriate permissions via the KMIP protocol.

The TOE supports the KMIP operations identified and described in **Table 4** below. Each operation has its own enabling permission. That is, to perform an operation successfully on a KMIP Object, the requesting KMIP User must be a member of a User Group that grants the equivalent permission to an Object Group containing the KMIP Object.

Operation	Description
Activate	Requests the TOE to activate a Managed Cryptographic Object
Add Attribute	Requests the TOE to add a new attribute instance to be associated with a Managed Object and to set its value
Archive	Specifies that a Managed Object MAY be archived
Cancel	Requests the TOE to cancel an outstanding asynchronous operation
Certify	Generate a Certificate object for a public key
Check	Requests the TOE to check for use of a Managed Object according to values specified in the request
Create	Requests the TOE to generate a new symmetric key as a Managed Cryptographic Object
Create Key Pair	Requests the TOE to generate a new public/private key pair and register the two corresponding new Managed Cryptographic Objects
Create Split Key	Requests the TOE to generate a new split key and register all the splits as individual new Managed Cryptographic Objects
Decrypt	Requests the TOE to perform a decryption operation on the provided data using a Managed Cryptographic Object as the key for the decryption operation
Delete Attribute	Requests the TOE to delete an attribute associated with a Managed Object
Derive Key	Requests the TOE to derive a symmetric key or secret Data object from a key or secret data that is already known to the TOE
Destroy	Used to indicate to the TOE that the key material for the specified Managed Object SHALL be destroyed
Encrypt	Requests the TOE to perform an encryption operation on the provided data using a Managed Cryptographic Object as the key for the encryption operation
Get	Requests that the TOE returns the Managed Object specified by its Unique Identifier
Get Attributes	Requests one or more attributes of a Managed Object
Get Attribute List	Requests a list of the attribute names associated with a Managed Object
Get Usage Allocation	Requests the TOE to obtain an allocation from the current Usage Limits value to allow the client to use the Managed Cryptographic Object for applying cryptographic protection
Hash	Requests the TOE to perform a hash operation on the data provided
Join Split Key	Requests the TOE to combine a list of Split Keys into a single Managed Cryptographic Object
Locate	Requests that the TOE search for one or more Managed Objects depending on the attributes specified in the request
MAC	Requests the TOE to perform a MAC operation on the provided data using a Managed Cryptographic Object as the key for the MAC operation
MAC Verify	Requests the TOE to perform a MAC verify operation on the provided data using a Managed Cryptographic Object as the key for the MAC verify operation

Operation	Description
Modify Attribute	Requests the TOE to modify the value of an existing attribute instance associated with a Managed Object
Obtain Lease	Requests the TOE to obtain a new Lease Time for a specified Managed Object
Poll	Requests the TOE to cancel an outstanding asynchronous operation
Recover	Used to obtain a Managed Object that has been archived
Register	Requests the TOE to register a Managed Object that was created by the client or obtained by the client through some other means, allowing the TOE to manage the object
Re-certify	Used to renew an existing certificate for the same key pair
Re-key	Used to generate a replacement key for an existing symmetric key
Re-key Key Pair	Used to generate a replacement key pair for an existing public/private key pair
Retrieve RNG	Requests the TOE to return output from a Random Number Generator
Revoke	Requests the TOE to revoke a Managed Cryptographic Object or an Opaque Object
Seed RNG	Requests the TOE to seed a Random Number Generator
Sign	Requests the TOE to perform a signature operation on the provided data using a Managed Cryptographic Object as the key for the signature operation
Signature Verify	Requests the TOE to perform a signature verify operation on the provided data using a Managed Cryptographic Object as the key for the signature verification operation
Validate	Requests the TOE validate a certificate chain and return information on its validity
Wrap	Requests the TOE to perform a key wrapping operation on the provided data using a Managed Cryptographic Object as the key for the wrapping operation

Table 4: Supported KMIP Operations

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1(1), FDP_ACF.1(1)—the TOE implements the ESKM SFP to control access of ESKM users to ESKM keys.
- FDP_ACC.1(2), FDP_ACF.1(2)—the TOE implements the KMIP SFP to control access of KMIP users to KMIP objects.
- FMT_MSA.1(1), FMT_MSA.3(1)—the TOE restricts the ability to manage ESKM key attributes to High Access Administrators and Administrators with the **Keys and Authorization Policies** privilege. High Access Administrators, Administrators with the **Keys and Authorization Policies** privilege, and the ESKM key owner can specify alternative initial values to override default values of security attributes when an ESKM key is created.
- FMT_MSA.1(2), FMT_MSA.3(2)—the TOE restricts the ability to manage KMIP key attributes to High Access Administrators and Administrators with the **Users and Groups** privilege. The KMIP user creating a KMIP object can specify alternative initial values to override default values of security attributes when a KMIP object is created.

6.1.4 Identification and Authentication

The TOE distinguishes between two types of user—administrators, who configure and manage the TOE, and clients, who request key management services of the TOE using KMS (ESKM users) or KMIP (KMIP users). The Identification and Authentication security function provides the capability for the TOE to identify and authenticate both administrators and clients.

6.1.4.1 Administrator I&A

In the evaluated configuration, administrator accounts are created and managed on the TOE (use of LDAP for remote identification and authentication is excluded from the evaluated configuration). The product also supports Public key authentication for CLI administration sessions however this feature has not been subjected to evaluation. Administrator usernames are restricted to letters and numbers only, must start with a letter, and can be up to 30 characters long. The TOE supports only passwords as the authentication credential for administrators. Administrator passwords must adhere to the TOE's password policies (see below).

When an administrator initiates a connection to the CLI (remotely via SSH) or the Management Console (remotely via HTTPS), the TOE prompts for the administrator username and associated password credential. The TOE uses the password to authenticate the claimed username by comparing it to the password value for that username stored on the TOE. If the TOE successfully authenticates the claimed administrator identity, the administrator is logged on to the TOE.

6.1.4.2 KMS Authentication

The TOE supports both passwords and certificates as mechanisms for KMS authentication. The following options are available in the evaluated configuration:

- Password only—the KMS server can be configured to require only a password for authentication of ESKM users
- Password and client certificate—the KMS server can be configured to require both a password and a client certificate for authentication of ESKM users.

Client certificate authentication in turn has two options:

- TLS session only—the ESKM user must provide a certificate signed by a CA trusted by the TOE in order to establish a TLS session with the KMS server
- TLS session and username—the ESKM user again must provide a certificate signed by a CA trusted by the TOE in order to establish a TLS session with the KMS server. In addition, a username is derived from the client certificate. The username can be derived from the UID (user ID), CN (Common Name), SN (Surname), E (Email address), E_ND (Email without domain), or OU (Organizational Unit) field, as specified in the KMS server configuration. The KMS server compares the username derived from the certificate with the username in the authentication request. If the usernames are the same and the password is valid, the user is authenticated. If the usernames are not the same, the connection is closed immediately.

When client certificate authentication is configured, the TOE verifies that the CA that signed the client certificate is in a list of Trusted CAs configured for the KMS server.

6.1.4.3 KMIP Authentication

The TOE supports the following KMIP user authentication mechanisms:

- Certificate-based authentication
- Authentication using credential objects (essentially username and password).

With certificate-based authentication, the KMIP user does not supply a credential structure in the KMIP client request. Instead, the client certificate used for TLS authentication is also used to determine the user identity. Regardless of the authentication mechanism used, a client certificate must be provided for TLS authentication.

An administrator with **Users and Groups** privilege must first add the KMIP-enabled user and specify the client certificate before a KMIP client can send a request using certificate authentication. The TOE stores the client certificate with the user properties in the KMIP user database. Unlike KMS, which can derive the username from the certificate by extracting it from fields such as the Common Name, KMIP does not require that the Common Name or any other field in the certificate match the username. Instead, the raw certificate contents, sent in the KMIP client request, are compared with the certificate contents configured in the KMIP user database, and if the values are the same the KMIP username is derived. Since the certificate contents are used to derive the username, the certificate must be unique. A single certificate cannot be shared by more than one KMIP-enabled user.

In addition to certificate-based authentication, the TOE supports KMIP client authentication using a Credential object. This is a structure used for client identification purposes and is managed by the TOE outside of KMIP. The Credential object contains two components:

- Credential Type—the TOE supports two types of credentials: "Username and Password"; and "Device"
- Credential Value.

If the Credential Type in the Credential object is "Username and Password", then the Credential Value in the KMIP client request will contain a username and password as text strings. For the authentication to succeed, the credential supplied in the KMIP client request must match the username and password that is configured in the TOE for the KMIP-enabled user.

If the Credential Type is "Device", the Credential Value is a structure that contains the password and some combination of the following components:

- Device serial number
- Device identifier
- Network identifier
- Machine identifier
- Media identifier.

The combination of these values must be unique.

For authentication using device credentials to succeed, the credential must match the username and password that is configured in the TOE for the KMIP-enabled user. The username must be of the following format:

device-serial-number:device-identifier:network-identifier:machine-identifier:media-identifier

For example, given the following values:

- device serial number = serial123
- device identifier = devid456
- network identifier = undefined (i.e. blank)
- machine identifier = machine1
- media identifier = undefined (i.e. blank)

The matching username must be configured on the TOE as follows:

serial123:devid456::machine1::

6.1.4.4 Password Controls

The TOE identifies four types of password, as follows:

- Administrator—used to authenticate an administrator's identity when logging on to the TOE
- Client—used by an ESKM User or KMIP User when connecting to the TOE to request key management services
- Backup—used to protect backup configurations
- Cluster—used to protect a cluster key.

All passwords on the TOE (i.e., administrator, client, backup, and cluster) are subject to the same basic constraints. Passwords must contain at least five different characters. Passwords must not:

- contain only whitespace
- resemble a phone number, dictionary word, or reversed dictionary word
- be based on the username associated with the password.

A High Access Administrator can configure the following additional constraints for all passwords:

- Minimum length (default is 8 characters)
- Minimum character requirements—the following can each be specified:
    - Must contain at least one lower case alphabetic character

- Must contain at least one upper case alphabetic character
- Must contain at least one numeric character
- Must contain at least one special character.

In addition, a High Access Administrator can configure the following constraints for administrator passwords:

- Password expiration—specifies the maximum number of days for which a password is valid. The maximum value that can be configured is 365 days. Once enabled, this feature applies to all current administrator passwords—all current administrator passwords have the same duration period, regardless of when they may have been created initially.
- Password history—specifies the number of passwords to maintain in the history. The acceptable range is from 1 to 25. When enabled, an administrator will not be able to reuse any password that is maintained in the administrator's password history.

High Access Administrators can change the password of any local administrator. If one administrator changes the password of another administrator, the administrator whose password changed is prompted to change his or her password immediately after logging in (with the new password) to the TOE. After changing the password, the administrator continues to the Management Console or the command prompt as usual.

6.1.4.5 Authentication Failure Handling

An administrator with **KMS/KMIP Server** privilege is able to configure the TOE to lockout the accounts of ESKM Users and KMIP Users to prevent a user from logging in to the TOE for a given duration after a specified number of failed login attempts. When the administrator enables this mechanism, the administrator specifies:

- the number of consecutive failed authentication attempts allowed before the account is locked (default is three)
- the duration (in seconds) of the lockout period (default is 60).

When the configured number of consecutive failed authentication attempts is met, the ESKM User or KMIP User account is locked for the configured number of seconds. During this time, the TOE will not accept any attempts to login to the account. Additionally, if SNMP is enabled an SNMP trap will be sent after a configurable number of failed login attempts.

6.1.4.6 User Attribute Definition

The Identification and Authentication security function maintains the following security attributes associated with users of the TOE:

- User identity—the user name the TOE uses to uniquely identify each user
- Authentication data—the credentials used to authenticate the user's identity
- Privileges—the administration privileges (termed "access controls" in the TOE guidance documentation) associated with the user.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1—the TOE is able to lock a user account after a configured number of consecutive failed attempts to logon.
- FIA_ATD.1—the TOE associates a user identity, authentication credentials, and administration privileges with each user.
- FIA_SOS.1—the TOE enforces minimum requirements for the construction of passwords.
- FIA_UAU.2—the TOE requires each user to be authenticated before gaining further access to TOE services.
- FIA_UAU.5—the TOE supports passwords and certificates as authentication mechanisms.
- FIA_UID.2—the TOE requires each user to be identified before gaining further access to TOE services.

6.1.5 Security Management

6.1.5.1 Security Management Roles

The TOE can support two types of administrators—local and LDAP. Functionally, local and LDAP administrators have the same capabilities. The difference is that local administrators are defined locally on the TOE appliance, while LDAP administrators are defined on an LDAP server in the operational environment. The definition and use of LDAP administrators is excluded from this evaluation.

Each administrator is assigned privileges (termed "access controls" in the TOE guidance documentation). A privilege is a permission to configure a TOE feature or perform an operation. Privileges are grouped into categories. The privileges defined by the TOE, their categories, and the permissions they grant, are summarized in the following table.

Category	Privilege	Permission
Security Configuration	Keys and Authorization Policies	Manage keys and Authorization Policies
	Users and Groups	Manage users and groups
	Certificates	Create and import certificates
	Certificate Authorities	Manage certificate authorities on the TOE
	Advanced Security	Manage advanced security settings
	SSL	Modify SSL configuration
Device Configuration	KMS/KMIP Server	Configure KMS and KMIP server settings
	Cluster	Manage cluster configuration
	Network and Date/Time	Configure network and date/time settings
	SNMP	Manage SNMP settings
	Logging	Manage logging settings
Backup & Restore	Backup Configuration	Create backups excluding keys, certificates, CAs
	Backup Keys and Certificates	Create backups of keys and certificates
	Backup Local CAs	Create backups of local CAs and associated private keys
	Restore Configuration	Restore backups excluding keys, certificates, CAs
	Restore Keys and Certificates	Restore backups of keys and certificates
	Restore Local CAs	Restore backups of local CAs and associated private keys
Maintenance	Services	Modify the startup service setting
	Software Upgrade and System Health	Upgrade to a new software version
Administrative Access	Admin Access via Web	Access the Management Console
	Admin Access via SSH	Access the CLI over SSH

Table 5: Administrator Privileges

The TOE includes a built-in administrator account ("admin"), which is allocated all privileges. An administrator account with all privileges assigned is identified as a High Access Administrator. A High Access Administrator can create other administrators and assign privileges to them. In addition, a High Access Administrator can revoke privileges assigned to an administrator. When this occurs, the revocation is enforced immediately—that is, the

administrator that has had a privilege revoked will not be able to perform the actions enabled by that privilege, even within an existing administrative session.

The TOE provides two interfaces by which administrative users can access and manage the TOE: the Management Console, a web-based GUI accessed using HTTPS; and a CLI accessFed using SSH. The TOE also provides local access to the CLI via a serial port, but this is intended for use only during initial installation and configuration of the TOE.

6.1.5.2 Security Management Functions and Management Restrictions

The Security Management security function provides the following capabilities for managing the behavior of the TSF and TSF data (capabilities for managing security attributes associated with the ESKM and KMIP access control policies are discussed in Section 6.1.3):

- Manage ESKM and KMIP Users—High Access Administrators and administrators with the **Users and Groups** privilege are able to create, modify and delete user accounts for ESKM and KMIP users
- Manage Administrators—High Access Administrators are able to create, modify, and delete administrator accounts
- Manage certificates—High Access Administrators and administrators with the **Certificates** privilege are able to create (including import) and delete certificates
- Manage Certificate Authorities—High Access Administrators and administrators with the **Certificate Authorities** privilege are able to create, modify and delete Certificate Authorities
- Manage connection timeout—High Access Administrators and administrators with the **KMS/KMIP Server** privilege are able to enable, disable and determine the behavior of the connection timeout function
- Manage password controls—High Access Administrators are able to modify the settings for password controls
- Manage audit security function—High Access Administrators and administrators with the **Logging** privilege are able to determine and modify the behavior of the audit function
- Manage cluster configuration—High Access Administrators and administrators with the **Cluster** privilege are able to create, modify and delete cluster configurations
- Change passwords—High Access Administrators are able to modify the passwords of other users (both administrators and clients)
- Manage NTP and date/time settings—High Access Administrators and administrators with the **NTP and Date/Time** privilege are able to modify the NTP settings and system date and time
- Manage User Account Lockout—High Access Administrators and administrators with the **KMS/KMIP Server** privilege are able to enable, disable and determine the behavior of user account lockout function
- Configure TLS—High Access Administrators and administrators with the **SSL** privilege are able to determine and modify the behavior of TLS
- Backup configurations—High Access Administrators and administrators with the **Backup Configuration**, **Backup Keys and Certificates**, and **Backup Local CAs** privilege are able to determine and modify the behavior of TOE backups
- Restore configurations—High Access Administrators and administrators with the **Restore Configuration**, **Restore Keys and Certificates**, and **Restore Local CAs** privilege are able to determine and modify the behavior of TOE restore operations.

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

- FMT_MOF.1(*)—the TOE is able to restrict the management of aspects of the TSF to administrators assigned specific privileges.

- FMT_MTD.1(*)—the TOE is able to restrict the management of TSF data to administrators assigned specific privileges.
- FMT_REV.1—the TOE provides the capability to immediately revoke privileges associated with administrator accounts.
- FMT_SMF.1—the TOE provides the capabilities necessary to manage the security of the TOE.
- FMT_SMR.1—the TOE defines security management roles based on the privileges assigned to individual administrator accounts. An administrator assigned all management privileges is termed a High Access Administrator.

6.1.6 TSF Protection

In its evaluated configuration, the TOE comprises two or more ESKM appliances configured as a single cluster, which provides redundancy and allows the TOE to continue to operate in a fully secure fashion in the event of a failure of a node in the cluster. Clustering also enables multiple ESKMs in a distributed environment to synchronize and replicate configuration information, which reduces administration overhead. Nodes in a cluster communicate with each other to maintain a synchronized configuration.

Communications between nodes in a cluster occur over TLS, which provides confidentiality and detection of modification of transmitted data. Confidentiality is provided through the use of AES or Triple DES, while detection of modification is provided through the use of HMAC-SHA-1, HMAC-SHA-256 or HMAC-SHA-384 (depending on the configured ciphersuite). In accordance with the TLS protocol, if the TOE detects an integrity error in a received TLS packet, it will generate a 'bad_record_mac' fatal alert and close down the connection.

When a configuration operation is performed on one cluster member, the cluster feature determines if the operation should be replicated throughout the cluster. If so, the device immediately sends a similar operation request to every other member using the cluster port (TCP port 9001 by default).

If the replication succeeds, the operation is recorded in the System Log. If the replication fails, the server waits 60 seconds and tries again. If 1440 consecutive replications fail, the server records the failure in the System Log and sends an SNMP trap indicating that the cluster is out of sync. Once a device is out of sync, an administrator must synchronize it manually.

The following configuration settings are replicated within a cluster:

- Keys
- Local Users and Groups
- KMS Server
- KMIP Server
- NTP
- DNS
- SNMP
- Log Signing Certificate
- Local Certificate Authorities (CAs)
- Authorization Policies
- TLS
- Administrators and Remote Administration
- Logging
- Service Startup
- Known CAs, CRLs, and Trusted CA List Profiles.

The TOE maintains time internally using a CMOS clock and this internal time is used as the source for the timestamp recorded in each audit record. An administrator with **Network and Date/Time** privilege can configure date/time settings on the TOE.

In addition, the TOE can be configured to synchronize its clock against one or more configured NTP servers. When the TOE attempts to synchronize its clock against an NTP server, one of three outcomes is possible:

- If the clock on the TOE is successfully synchronized, and the difference between the time on the TOE and the NTP server is less than 0.5 seconds, the time on the TOE is gradually slewed to the real time.
- If the clock on the TOE is successfully synchronized, and the difference between the time on the TOE and the NTP server is greater than 0.5 seconds, the time on the TOE is immediately stepped to the real time. This event is recorded in the System Log.
- If an error prevented the TOE from synchronizing its clock, an error message is recorded in the System Log.

An administrator with **Network and Date/Time** privilege can configure the TOE to use NTP.

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

- FPT_FLS.1—the TOE is able to preserve a secure state in the event a node in the cluster fails.
- FPT_ITT.1—the TOE uses TLS to protect TSF data from disclosure when it is transmitted between nodes in the cluster.
- FPT_ITT.3—the TOE uses TLS to detect modification of TSF data transmitted between nodes in the cluster. If the TOE detects data modification, it will close down its connection with the other node.
- FPT_STM.1—the TOE maintains its own internal clock to provide reliable time stamps and can also be configured to synchronize its clock with NTP servers.

6.1.7 TOE Access

An administrator can configure the TOE to terminate the following session types after a specified period of inactivity:

- KMS Server—the connection timeout value specifies in seconds how long ESKM client connections can remain idle before the KMS Server begins closing them. The default value is 3600 seconds (1 hour); the maximum value is 7200 seconds (2 hours). A value of 0 means that the KMS Server will not close client connections due to inactivity. The administrator requires the **KMS/KMIP Server** privilege to configure this setting
- KMIP Server—similarly, the connection timeout value specifies in seconds how long KMIP client connections can remain idle before the KMIP Server begins closing them. The default value is 3600 seconds (1 hour); the maximum value is 7200 seconds (2 hours). A value of 0 means that the KMIP Server will not close client connections due to inactivity. The administrator requires the **KMS/KMIP Server** privilege to configure this setting
- CLI—the administrator uses the `autologout` CLI command to set the number of minutes the CLI remains inactive prior to logging off the current user. The administrator can specify a timeout value from 1 to 720 minutes. The default is 30 minutes. A value of 0 disables autologout.
- Management Console—the inactivity timeout value for Management Console sessions is fixed at 60 minutes.

The TOE allows administrators to terminate their own interactive sessions. At the CLI, the administrator enters the `exit` command while in `view` mode in order to terminate the interactive session. At the Management Console, the administrator clicks on the 'Log Out' link at the top right portion of the GUI to terminate the interactive session and return to the GUI login screen.

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

- FTA_SSL.3—the TOE will terminate interactive sessions after a period of inactivity configurable by an administrator.
- FTA_SSL.4—the TOE allows administrators to terminate their own interactive sessions.

6.1.8 Trusted Channel/Path

6.1.8.1 Trusted Channel

The TOE supports communications via trusted channels with other trusted IT products for the following functions:

- Key management services
- TOE backup and restore.

ESKM clients (i.e., encrypting client devices and applications) can initiate communication via the trusted channel to submit requests for key management services using the ESKM XML protocol or KMIP. In the evaluated configuration, this channel is implemented using TLS 1.0 or higher (the TOE implements TLS 1.0, 1.1 and 1.2). The TOE supports the following TLS ciphersuites, as defined in RFC 2246 and RFC 4346:

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_TDES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384.

The TOE can be configured to create backups of its configuration to assist in recovering the TOE in the event of a TOE failure or inter-TOE communications failure, which could cause nodes in the TOE cluster to become out-of-sync. An administrator with one of the backup privileges (**Backup Configuration**, **Backup Keys and Certificates**, **Backup Local CAs**) can specify the backup file is to be exported to a trusted IT product using SCP (Secure Copy). SCP is based on SSH. The TOE implements SSH v2using 2048-bit RSA keys for digital signature generation and verification, 168-bit Triple DES, or 128, 192, or 256-bit AES keys for session data encryption and decryption, and HMAC-SHA-512 for data authentication.

Similarly, an administrator with the appropriate privilege (**Restore Configuration**, **Restore Keys and Certificates**, **Restore Local CAs**) can use SCP to import a backup file from a trusted IT product and restore the TOE's configuration, keys, certificates and local CAs.

6.1.8.2 Trusted Path

The TOE provides a trusted path for administrators of the TOE to communicate with the TOE. The trusted path is implemented using HTTPS (i.e., TLS over HTTP) for access to the Management Console and SSHv2 for access to the CLI. Administrators initiate the trusted path by establishing an HTTPS connection (using a supported web browser) or SSH session (using an SSH client). The TOE's implementation of TLS and SSH are as described in the previous section (Trusted Channel).

The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS or SSHv2 ensures all communication over the trusted path is protected from disclosure and undetected modification.

The Trusted Channel/Path function is designed to satisfy the following security functional requirements:

- FTP_ITC.1—the TOE supports establishment of trusted channels for communicating with trusted IT entities using TLS.
- FTP_TRP.1—the TOE provides a trusted path for administrators to communicate with the TOE, using SSHv2 to access the CLI and HTTPS to access the Management Console.

7. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

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

7.1 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. In addition, each objective counters or addresses at least one assumption or threat.

	T.BRUTE_FORCE	T.DATA_COMPROMISE	T.KEY_COMPROMISE	T.NETWORK_COMPROMISE	T.NO_ACCOUNTABILITY	T.UNATTENDED_SESSION	T.UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_ACTIVITY	T.UNAVAILABILITY	T.UNRECOVERABLE_DATA	A.PROTECT	A.MANAGE	A.NOEVIL
O.AUDIT					X								
O.AUDIT_REVIEW					X								
O.AUDIT_STORAGE					X								
O.CRYPTOGRAPHY				X									
O.HIGH_AVAILABILITY									X				
O.I_&_A							X						
O.KEY_ACCESS			X										
O.KEY_MANAGEMENT		X								X			
O.PASSWORD_CONTROLS	X												
O.PROTECTED_COMMS				X									
O.SECURITY_MANAGEMENT								X					
O.SESSION_TERMINATION						X							
O.THROTTLE	X												
OE.PHYSICAL											X		
OE.PERSONNEL												X	X

Table 6: Security Problem Definition to Security Objective Correspondence

7.1.1.1 T.BRUTE_FORCE

An unauthorized user may gain access to the TOE through repeated password-guessing attempts.

This threat is countered by the following security objectives:

- O.PASSWORD_CONTROLS—addresses this threat by providing a mechanism, configurable by an administrator, which encourages users to choose difficult-to-guess passwords.

- O.THROTTLE—addresses this threat by limiting the number of passwords that can be guessed for a single account to a rate of six per minute.

7.1.1.2 T.DATA_COMPROMISE

Data on long-term storage media may be compromised if control of that media passes to unauthorized entities.

This threat is countered by the following security objective:

- O.KEY_MANAGEMENT—addresses this threat by providing services that clients can request for generating and managing encryption keys that can be used by the clients to encrypt data to be stored long-term.

7.1.1.3 T.KEY_COMPROMISE

Encrypted data may be compromised if unauthorized users gain access to encryption keys.

This threat is countered by the following security objective:

- O.KEY_ACCESS—addresses this threat by ensuring access to keys managed by the TOE is restricted to authorized users.

7.1.1.4 T.NETWORK_COMPROMISE

TSF data communicated between components of the TOE, or between the TOE and external entities, is disclosed or undetectably modified.

This threat is countered by the following security objectives:

- O.PROTECTED_COMMS—addresses this threat by ensuring all communications between distributed parts of the TOE, and between the TOE and external entities, are protected using cryptographic protocols, such as SSH and TLS.
- O.CRYPTOGRAPHY—supports O.PROTECTED_COMMS by ensuring the TOE implements the cryptographic algorithms necessary to support the cryptographic protocols that satisfy O.PROTECTED_COMMS.

7.1.1.5 T.NO_ACCOUNTABILITY

Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.

This threat is countered by the following security objectives:

- O.AUDIT—addresses this threat by ensuring the TOE is able to generate audit records of security relevant events.
- O.AUDIT_REVIEW—supports O.AUDIT in addressing the threat by ensuring the TOE provides capabilities for effective review of stored audit records.
- O.AUDIT_STORAGE—supports O.AUDIT in addressing the threat by ensuring the TOE protects stored audit records from unauthorized modification and deletion.

7.1.1.6 T.UNATTENDED_SESSION

An unauthorized user gains access to the TOE via an unattended authorized user session.

This threat is countered by the following security objectives:

- O.SESSION_TERMINATION—addresses this threat by providing users with a mechanism to terminate their interactive sessions with the TOE, and by ensuring sessions that have been inactive for a configurable period of time will be terminated by the TOE.

7.1.1.7 T.UNAUTHORIZED_ACCESS

Unauthorized users gain access to the TOE and its services.

This threat is countered by the following security objective:

- O.I_AND_A—addresses this threat by ensuring all users of the TOE are identified and authenticated prior to gaining further access to the TOE and its services.

7.1.1.8 T.UNAUTHORIZED_ACTIVITY

Authorized users perform unauthorized actions on the TOE.

This threat is countered by the following security objective:

- O.SECURITY_MANAGEMENT—addresses this threat by providing a mechanism that requires authorized users to have appropriate privileges in order to perform actions on the TOE.

7.1.1.9 T.UNAVAILABILITY

Authorized users are unable to access TOE services due to failure of the TOE.

This threat is countered by the following security objective:

- O.HIGH_AVAILABILITY—addresses this threat by ensuring the TOE is able to continue serving user requests after a failure of part of the TOE.

7.1.1.10 T.UNRECOVERABLE_DATA

Encrypted data may be unrecoverable if encryption keys are mishandled.

This threat is countered by the following security objective:

- O.KEY_MANAGEMENT—addresses this threat by providing services for securely storing and managing encryption keys used by clients to encrypt data.

7.1.1.11 A.PROTECT

The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

This assumption is satisfied by the following security objectives:

- OE.PHYSICAL—this objective satisfies the assumption by ensuring the TOE is protected from physical attack.

7.1.1.12 A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This assumption is satisfied by the following security objectives:

- OE.PERSONNEL—this objective satisfies the assumption by ensuring those assigned as authorized administrators are properly trained in operating the TOE.

7.1.1.13 A.NOEVIL

The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

This assumption is satisfied by the following security objectives:

- OE.PERSONNEL—this objective satisfies the assumption by ensuring those assigned as authorized administrators are carefully selected for the job and are properly trained in operating the TOE.

7.2 Security Functional Requirements Rationale

All security functional requirements identified in this Security Target are fully addressed in this section and each is mapped to the objective it is intended to satisfy. **Table 7** summarizes the correspondence of functional requirements to TOE security objectives.

	O.AUDIT	O.AUDIT_REVIEW	O.AUDIT_STORAGE	O.CRYPTOGRAPHY	O.HIGH_AVAILABILITY	O.I_AND_A	O.KEY_ACCESS	O.KEY_MANAGEMENT	O.PASSWORD_CONTROLS	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SESSION_TERMINATION	O.THROTTLE
FAU_GEN.1	X												
FAU_GEN.2	X												
FAU_SAR.1		X											
FAU_STG.1			X										
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						
FIA_AFL.1													X
FIA_ATD.1						X							
FIA_SOS.1									X				
FIA_UAU.2						X							
FIA_UAU.5						X							
FIA_UID.2						X							
FMT_MOF.1(*)											X		
FMT_MSA.1(*)							X						
FMT_MSA.3(*)							X						
FMT_MTD.1(*)											X		
FMT_SMF.1											X		
FMT_SMR.1											X		
FMT_REV.1											X		
FPT_FLS.1					X								
FPT_ITT.1										X			
FPT_ITT.3										X			
FPT_STM.1	X												
FTA_SSL.3												X	
FTA_SSL.4												X	
FTP_ITC.1										X			
FTP_TRP.1										X			

Table 7: Objectives to Requirement Correspondence

7.2.1.1 O.AUDIT

The TOE shall be able to generate audit records of security-relevant events, identifying users causing the events as applicable.

The following security functional requirements contribute to satisfying this security objective:

- FAU_GEN.1—the ST includes FAU_GEN.1 to specify the capability to generate audit records of security-relevant events, and to specify the specific events to be audited and the content of generated audit records of those events.
- FAU_GEN.2—the ST supports FAU_GEN.1 by including FAU_GEN.2 to specify the capability to include, when applicable, the identity of the user associated with the auditable event.
- FPT_STM.1—the ST supports FAU_GEN.1 by including FPT_STM.1 to specify the capability for the TOE to provide reliable time stamps, which are used by the TOE when generating audit records

7.2.1.2 O.AUDIT_REVIEW

The TOE shall provide a means for authorized users to review the audit records generated by the TOE.

The following security functional requirement contributes to satisfying this security objective:

- FAU_SAR.1—the ST includes FAU_SAR.1 to specify which roles are to be able to read data from stored audit records.

7.2.1.3 O.AUDIT_STORAGE

The TOE shall protect stored audit records from unauthorized modification and deletion.

The following security functional requirement contributes to satisfying this security objective:

- FAU_STG.1—the ST includes FAU_STG.1 to specify that stored audit records are to be protected from unauthorized modification or deletion.

7.2.1.4 O.CRYPTOGRAPHY

The TOE shall perform cryptographic operations to support protocols used to protect data in transit.

The following security functional requirements contribute to satisfying this security objective:

- FCS_COP.1(*)—the ST includes iterations FCS_COP.1 to specify the cryptographic algorithms implemented by the TOE to support cryptographic protocols.

7.2.1.5 O.HIGH_AVAILABILITY

The TOE shall provide the capability to continue servicing user requests after a failure of part of the TOE.

The following security functional requirement contributes to satisfying this security objective:

- FPT_FLS.1—the ST includes FPT_FLS.1 to specify the capability to maintain a secure state in the event of a failure of part of the TOE.

7.2.1.6 O.I_AND_A

The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.

The following security functional requirements contribute to satisfying this security objective:

- FIA_UID.2, FIA_UAU.2—the ST includes FIA_UID.2 and FIA_UAU.2 to specify that users must be successfully identified and authenticated by the TOE before being able to perform any other TSF-mediated actions.
- FIA_ATD.1—the ST supports FIA_UID.2 and FIA_UAU.2 by including FIA_ATD.1 to ensure user identity and authentication data security attributes are associated with individual users.

- FIA_UAU.5—the ST supports FIA_UAU.2 by including FIA_UAU.5 to specify multiple authentication mechanisms that are supported by the TOE.

7.2.1.7 O.KEY_ACCESS

The TOE shall restrict access to managed encryption keys to authorized users.

The following security functional requirements contribute to satisfying this security objective:

- FDP_ACC.1(1), FDP_ACF.1(1)—the ST includes FDP_ACC.1(1) and FDP_ACF.1(1) to specify the access control policy enforced by the TOE to control access to KMS objects to authorized KMS users.
- FMT_MSA.1(1), FMT_MSA.3(1)—the ST includes FMT_MSA.1(1) and FMT_MSA.3(1) to specify restrictions on the management of security attributes used to control access to KMS objects.
- FDP_ACC.1(2), FDP_ACF.1(2)—the ST includes FDP_ACC.1(2) and FDP_ACF.1(2) to specify the access control policy enforced by the TOE to control access to KMIP objects to authorized KMIP users.
- FMT_MSA.1(2), FMT_MSA.3(2)—the ST includes FMT_MSA.1(2) and FMT_MSA.3(2) to specify restrictions on the management of security attributes used to control access to KMIP objects.

7.2.1.8 O.KEY_MANAGEMENT

The TOE shall provide services for authorized users to request generation of encryption keys and shall provide management services for those keys.

The following security functional requirements contribute to satisfying this security objective:

- FCS_CKM.1—the ST includes FCS_CKM.1 to specify the capability to generate symmetric cryptographic keys and asymmetric key pairs at the request of authorized users.
- FCS_CKM.2—the ST includes FCS_CKM.2 to specify the capability to distribute cryptographic keys to authorized users.
- FCS_CKM.3—the ST includes FCS_CKM.3 to specify the capability to store cryptographic keys on behalf of authorized users.
- FCS_CKM.4—the ST includes FCS_CKM.4 to specify the capability to destroy cryptographic keys at the request of authorized users.

7.2.1.9 O.PASSWORD_CONTROLS

The TOE shall provide a mechanism to reduce the likelihood that users choose weak passwords.

The following security functional requirements contribute to satisfying this security objective:

- FIA_SOS.1—the ST includes FIA_SOS.1 to specify that passwords must meet minimum construction requirements, in terms of length and character set.

7.2.1.10 O.PROTECTED_COMMS

The TOE shall protect communications between distributed parts of the TOE, and between the TOE and external entities, from disclosure and undetected modification.

The following security functional requirements contribute to satisfying this security objective:

- FPT_ITT.1, FPT_ITT.3—the ST includes FPT_ITT.1 and FPT_ITT.3 to specify that communications between distributed parts of the TOE will be protected from disclosure and undetected modification.
- FTP_ITC.1, FTP_TRP.1—the ST includes FTP_ITC.1 and FTP_TRP.1 to specify that communications between the TOE and external entities will be protected from disclosure and undetected modification.

7.2.1.11 O.SECURITY_MANAGEMENT

The TOE shall restrict the ability to perform security management functions on the TOE to authorized administrators having appropriate privileges.

The following security functional requirements contribute to satisfying this security objective:

- FMT_SMF.1, FMT_SMR.1, FMT_MOF.1(*), FMT_MTD.1(*), FMT_REV.1—the ST includes these requirements to specify the security management functions to be provided by the TOE (FMT_SMF.1), to specify security management roles and privileges (FMT_SMR.1), to specify the restrictions on management of security function behavior and TSF data (FMT_MOF.1(*), FMT_MTD.1(*)), and to specify TOE behavior when security management privileges are revoked (FMT_REV.1).

7.2.1.12 O.SESSION_TERMINATION

The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.

The following security functional requirements contribute to satisfying this security objective:

- FTA_SSL.3—the ST includes FTA_SSL.3 to specify the capability for the TSF to terminate an interactive user session after a period of inactivity.
- FTA_SSL.4—the ST includes FTA_SSL.4 to specify the capability for users to terminate their own interactive sessions.

7.2.1.13 O.THROTTLE

The TOE shall limit the rate at which consecutive unsuccessful authentication attempts can be performed.

The following security functional requirement contributes to satisfying this security objective:

- FIA_AFL.1—the ST includes FIA_AFL.1 to specify the capability to limit the rate at which consecutive failed authentication attempts (which may indicate a password-guessing attack) can be made.

7.3 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 will have Basic attack potential. Augmentation was chosen to provide the added assurance that is gained by defining flaw remediation and flaw reporting procedures. Therefore, the target assurance level of EAL 2 augmented with ALC_FLR.2 is appropriate for such an environment.

7.4 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. It can be seen that all dependencies have been satisfied by inclusion in the ST of the appropriate dependent SFRs.

Requirement	Dependencies	How Satisfied
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1,
	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]	FCS_CKM.2 and FCS_COP.1
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.3	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1

Requirement	Dependencies	How Satisfied
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
	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.3	FMT_MSA.3(*)
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2 (hierarchical to FIA_UAU.1)
FIA_ATD.1	None	n/a
FIA_SOS.1	None	n/a
FIA_UAU.2	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FIA_UAU.5	None	n/a
FIA_UID.2	None	n/a
FMT_MOF.1(*)	FMT_SMF.1	FMT_SMF.1
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.1(*)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(*)
	FMT_SMF.1	FMT_SMF.1
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(*)	FMT_MSA.1	FMT_MSA.1(*)
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1(*)	FMT_SMF.1	FMT_SMF.1
	FMT_SMR.1	FMT_SMR.1
FMT_REV.1	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	None	n/a
FMT_SMR.1	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FPT_FLS.1	None	n/a
FPT_ITT.1	None	n/a
FPT_ITT.3	FPT_ITT.1	FPT_ITT.1
FPT_STM.1	None	n/a
FTA_SSL.3	None	n/a
FTA_SSL.4	None	n/a
FTP_ITC.1	None	n/a
FTP_TRP.1	None	n/a

Table 8: Requirement Dependencies

7.5 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

	Security Audit	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	TSF Protection	TOE Access	Trusted Path
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAR.1	X							
FAU_STG.1	X							
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					
FIA_AFL.1				X				
FIA_ATD.1				X				
FIA_SOS.1				X				
FIA_UAU.2				X				
FIA_UAU.5				X				
FIA_UID.2				X				
FMT_MOF.1					X			
FMT_MSA.1(*)			X					
FMT_MSA.3(*)			X					
FMT_MTD.1(*)					X			
FMT_REV.1					X			
FMT_SMF.1					X			
FMT_SMR.1					X			
FPT_FLS.1						X		
FPT_ITT.1						X		
FPT_ITT.3						X		
FPT_STM.1						X		
FTA_SSL.3							X	
FTA_SSL.4							X	
FTP_ITC.1								X
FTP_TRP.1								X

Table 9: Security Functions vs. Requirements Mapping