ERUCES Tricryption KeyServer and Agent 6.2

Security Target

Version 0.8

08/27/2009

Prepared for:

ERUCES, Inc.
11142 Thompson Ave.
Lenexa, KS 66219-2301

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

1. SECURITY TARGET INTRODUCTION 4
1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 4
1.2 CONFORMANCE CLAIMS 4
1.3 CONVENTIONS 5
2. TOE DESCRIPTION 6
2.1 TOE OVERVIEW 6
2.2 TOE ARCHITECTURE 6
2.2.1 Physical Boundaries 7
2.2.2 Logical Boundaries 8
2.3 TOE DOCUMENTATION 9
3. SECURITY ENVIRONMENT 10
3.1 ORGANIZATION SECURITY POLICIES 10
3.2 THREATS 10
3.3 ASSUMPTIONS 10
4. SECURITY OBJECTIVES 12
4.1 SECURITY OBJECTIVES FOR THE TOE 12
4.2 SECURITY OBJECTIVES FOR THE IT ENVIRONMENT 12
4.3 SECURITY OBJECTIVES FOR THE NON-IT ENVIRONMENT 12
5. IT SECURITY REQUIREMENTS 14
5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 14
5.1.1 Security audit (FAU) 14
5.1.2 Cryptographic Support (FCS) 15
5.1.3 User Data Protection (FDP) 16
5.1.4 Identification and Authentication (FIA) 17
5.1.5 Security management (FMT) 17
5.1.6 ***Protection of the TSF (FPT)*** *....18*
5.1.7 TOE Access (FTA) 18
5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS 18
5.2.1 ***Protection of the TSF (FPT)*** *....19*
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 19
5.3.1 Configuration Management (ACM) 20
5.3.2 Delivery and operation (ADO) 20
5.3.3 Development (ADV) 20
5.3.4 Guidance documents (AGD) 21
5.3.5 Life cycle (ALC) 22
5.3.6 Tests (ATE) 22
5.3.7 Vulnerability assessment (AVA) 23
6. TOE SUMMARY SPECIFICATION 25
6.1 TOE SECURITY FUNCTIONS 25
6.1.1 Security Audit 25
6.1.2 Cryptographic Protection 26
6.1.3 User Data Protection 26
6.1.4 Identification and Authentication 27
6.1.5 Security Management 28
6.1.6 TOE Access 29
6.2 TOE SECURITY ASSURANCE MEASURES 30
6.2.1 Configuration management 30
6.2.2 Delivery and operation 30
6.2.3 Development 30

6.2.4 Guidance documents 31
6.2.5 Life cycle support 31
6.2.6 Tests 31
6.2.7 Vulnerability assessment 32

7. PROTECTION PROFILE CLAIMS 32

8. RATIONALE 33

8.1 SECURITY OBJECTIVES RATIONALE 33
8.1.1 Security Objectives Rationale for the TOE and Environment 33
8.2 SECURITY FUNCTIONAL REQUIREMENTS RATIONALE 37
8.2.1 Security Functional Requirements Rationale 37
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE 41
8.4 STRENGTH OF FUNCTIONS RATIONALE 42
8.5 REQUIREMENT DEPENDENCY RATIONALE 42
8.6 EXPLICITLY STATED REQUIREMENTS RATIONALE 43
8.7 TOE SUMMARY SPECIFICATION RATIONALE 43
8.8 PP CLAIMS RATIONALE 44

LIST OF TABLES

Table 1 – SFRs satisfied by the TOE 14
Table 2 – Cryptographic key operations 15
Table 3 – IT Environment Security Functional Components 19
Table 4 – EAL 2 and ALC_FLR.2, and AVA_MSU.1 assurance components 19
Table 5 – Environment to Objective Correspondence 33
Table 6 – Objective to Requirement Correspondence 38
Table 7 – Dependency Analysis 42
Table 8 – Security Functions vs. Requirements Mapping 44

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Tricryption KeyServer and Agent 6.2, provided by ERUCES, Inc.

The Security Target contains the following additional sections:

- TOE Description (Section 2)

  This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.

- Security Environment (Section 3)

  This section details the expectations of the environment, the threats that are countered by the TOE and it environment and the organizational policy that the TOE must fulfill.

- Security Objectives (Section 4)

  This section details the security objectives of the TOE and its environment.

- IT Security Requirements (Section 5)

  The section presents the security functional requirements (SFR) for TOE and IT Environment that supports the TOE, and details the assurance requirements for EAL 2, ALC_FLR.2, and AVA_MSU.1.

- TOE Summary Specification (Section 6)

  The section describes the security functions represented in the TOE that satisfy the security requirements

- Protection Profile Claims (Section 7)

  This section presents any protection profile claims

- Rationale (Section 8).

  This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – ERUCES Tricryption KeyServer and Agent 6.2 Security Target

ST Version – Version 0.8

ST Date – 08/27/2009

TOE Identification – Tricryption KeyServer and Agent 6.2

TOE Developer – ERUCES, Inc.

Evaluation Sponsor – ERUCES, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.

  - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - EAL 2+: Evaluation Assurance Level 2, ALC_FLR.2 (Flaw Remediation)
  - Strength of Function Claim: SOF-Basic

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 the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**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 to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is the ERUCES Tricryption KeyServer and Tricryption Agent 6.2.

The TOE includes server application components that perform cryptographic and key management operations, which are collectively called Tricryption. The TOE also includes Application Programming Interfaces (APIs) that can be called by applications outside of the TOE boundary to access TOE server services. Note that the TOE does not store application data that has been encrypted using the TOE.

2.1 TOE Overview

The TOE provides APIs to applications outside of the TOE boundary that allow application users to share encrypted application data. The TOE provides the ability to share encrypted application data by providing cryptographic and key management operations in its server-side components.

2.2 TOE Architecture

The TOE provides the discretionary access control protocol using roles and access control lists (ACLs) associated with the cryptographic keys used to encrypt and decrypt data. Encrypted data will have an associated cryptographic key that must be used to decrypt the item. The owner of the encrypted data can authorize access to the key needed to decrypt the data. Only a user with the requested role and allowed access to a data's cryptographic key may decrypt the data.

The TOE in its intended environment can be described in terms of the following components:

- TOE Components
    - Tricryption KeyServer application server component (Keyserver) – Provides Tricryption services that are accessible using network protocol interfaces.
    - Tricryption Agent API component (Agent) – Provides programmatic interfaces to Tricryption KeyServer subcomponent services.
        - Tricryption Agent COM subcomponent – Microsoft COM DLL implementation of Agent API.
        - Tricryption Agent Java subcomponent – Java implementation of Agent API.
        - Tricryption Agent C++ subcomponent – C++ implementation of Agent API.
        - Tricryption Agent C subcomponent – C implementation of Agent API.
    - Application that calls Tricryption Agent API component – Requests Tricryption KeyServer services using Agent API.
    - Tricryption KeyServer administrator console component – Provides interfaces that can be used to manage TOE security functions.
        - Tricryption Manager, subcomponent – Provides a graphical user interface (GUI) application that can be used to manage the Tricryption KeyServer application server components.
        - Tricryption LogReporter, subcomponent – Provides a graphical user interface (GUI) application that can be used for authorized administrator to view the Tricryption KeyServer audit logs.
    - Tricryption Cryptographic Module – Provides cryptographic operations supporting the other Tricryption operations, as well as certificate authentication mechanism operations and SSL/TLS capabilities.

- Environment Components
    - Operating system – Provides runtime environment for Tricryption KeyServer application server components and JVM.
    - Database – Provides storage for encrypted cryptographic keys used by the TOE.
    - Java Virtual Machine (JVM) – Provides runtime environment for Tricryption Manager subcomponent.
    - Certification Authority (CA) – Provides digital certificates for Tricryption KeyServer application server subcomponents.

During installation the Tricryption KeyServer is configured with its own asymmetric key pair and corresponding certificate, and a user account is created (using Tricryption administrator console). The key pair and certificate is stored in operating system files on the Tricryption KeyServer machine in the IT environment. The Keyserver is also configured with a set of system keys which are stored in the database in the IT environment.

When an application outside of the TOE boundary calls the Tricryption API to encrypt data, the Tricryption API establishes an SSL connection with the Tricryption KeyServer. The Tricryption KeyServer presents its certificate to establish the SSL connection; the application outside of the TOE boundary in turn presents a username and password via the Tricryption API to the Tricryption KeyServer after the SSL connection is established.

When the Tricryption KeyServer receives a request to encrypt data, it creates a new symmetric key called a “session key” to encrypt the data, as well as creating a new identifier that corresponds to the encrypted version of the key, as well as creating a new access control list associated with the identifier. The access control list can then be used together with the key identifier by the Tricryption KeyServer to control access to keys generated to encrypt data to individual users. After the key, identifier, and ACL are created, the data is then encrypted by the Tricryption KeyServer using the session key and the key identifier is encrypted using one of one of the system keys. Both the encrypted data and the encrypted key identifier are returned to the application outside the TOE boundary via the Tricryption API using the already-established SSL connection. The session key is then encrypted using the system key. The encrypted key id becomes a hidden link between the encrypted data and the store encrypted session key. Note that the session key, encrypted key id and system keys are all store in the database in the IT environment.

When the Tricryption KeyServer receives a request to decrypt data, the application outside of the boundary first establishes an SSL connection that is authenticated in the same way as described above, also using the Tricryption API. The client then sends the encrypted data and its encrypted identifier to the server. The server decrypts (using the system key) the encrypted identifier, then uses the unencrypted identifier as a database key to retrieve from the database the corresponding encrypted session key. The server then checks the unencrypted identifier against the encrypted session key ACL. If the access control check is successful, the encrypted session key is then decrypted (using the system key), and then the encrypted data is decrypted using the decrypted session key. The decrypted data is then returned to the application outside of the TOE boundary via the Tricryption API using the already-established SSL connection.

2.2.1 Physical Boundaries

The components that make up the TOE are:

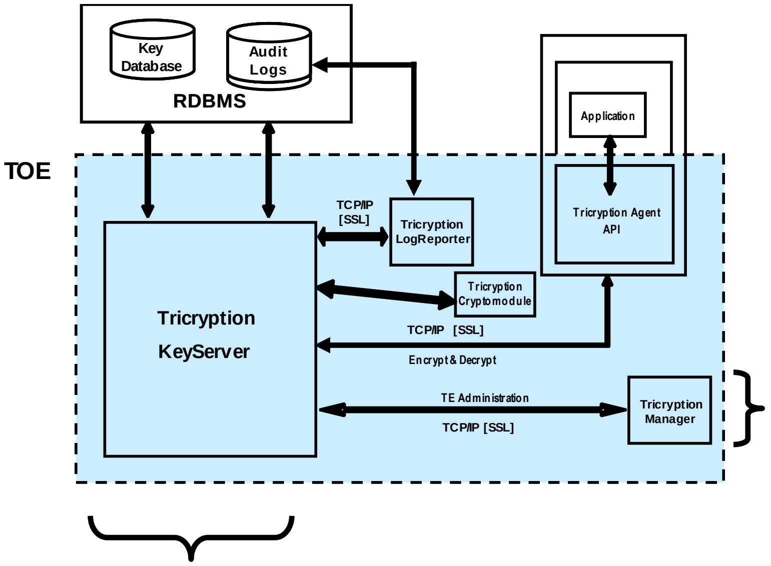
- Tricryption KeyServer application server component
- Tricryption Agent API component
- Tricryption administrator console component
- Tricryption Cryptographic Module

The TOE depends on the following:

- Operating system – Microsoft Windows 2000, 2003 or XP, RedHat Linux with kernel version 2.4, 2.6.
- Database – Microsoft SQL Server, MySQL MySQL v4.x
- Java Virtual Machine (JVM) – Java 2 Runtime Environment, Version 1.5

- Certification Authority (CA) – Those that support base 64-encoded PKCS#10/7 certificate requests/responses.

The TOE in its intended environment is depicted in the figure below. Note that the Tricryption Manager, Tricryption LogReporter and Tricryption Agent API , which is used by applications, can be installed either on the same machine as the Tricryption KeyServer or on a different machine or on a different platform. The Tricryption Cryptographic Module is installed as an option of the KeyServer installation and is always on the same machine.



Manager component runs in JVM which runs in context of operating system

Evaluated configurations include those with multiple instances of the engine subcomponent.

No network interfaces to CAs are supported, requests/responses use base-64-encoded PKCS#10/7 messages written to files.

Tricryption Manager, Tricryption LogReporter and Tricryption Agent API can be installed either on the same machine as the Tricryption KeyServer or on a different machine or on a different platform

Figure 1 - TOE Boundary

2.2.2 Logical Boundaries

The TOE logically supports the following security functions at its interfaces:

- Security Auditing,
- Cryptographic Protection,
- User Data Protection,
- Identification and Authentication,
- Security Management, and
- TOE Access.

Note that the TOE relies on its environment for its own protection. First, the TOE relies on its underlying operating system(s) to ensure that TOE applications and data stores cannot be accessed in any manner that might allow the TOE to be tampered with or bypassed.

2.2.2.1 Security Auditing

The TOE generates audit events for a minimum level of audit. Access control and management events are audited. The audit trail is stored in a database in the IT Environment.

2.2.2.2 Cryptographic Protection

The TOE includes a FIPS 140-2 certified cryptographic module (Tricryprtion Cryptomodule) that is used to generate and destroy cryptographic keys as well as to implement encryption/decryption functions that are used to support the other security functions and also to facilitate secure (i.e., SSL/TLS) communication among TOE components.

2.2.2.3 User Data Protection

The TOE can control access by user to encrypted keys using ACLs. Encrypted keys can have non-administrative owners. Tricryption API component interfaces can be used by encrypted key owners to access encrypted keys, including managing corresponding ACLs.

2.2.2.4 Identification and Authentication

The TOE provides a username/password Secure Remote Password (SRP) (http://srp.stanford.edu/) authentication mechanism. The TOE also provides a certificate authentication mechanism that relies on the cryptomodule API in the IT environment in order to operate, to use for inter-TSF component authentication.

2.2.2.5 Security Management

The TOE supports user-defined groups and pre-defined roles. The Tricryption administrator console component provides interfaces to manage user-defined groups and user-defined roles, as well as to manage TOE security functions.

2.2.2.6 TOE Access

The TOE can terminate Tricryption administrator console component sessions after an administrator-configured (using the Tricryption administrator console) period of time.

2.3 TOE Documentation

ERUCES offers a series of documents that describe the installation process for the Tricryption KeyServer and Agent 6.2 as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documentation associated with the Tricryption KeyServer and Agent 6.2.

3. Security Environment

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed.

The statement of the TOE security environment defines the following:

- Organizational security policies which the TOE is designed to comply.
- Threats that the TOE is designed to counter
- Assumptions made on the operational environment and the method of use intended for the TOE

3.1 Organization Security Policies

P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their actions within the TOE.

P.AUTHORIZATION

The abilities of users of the TOE shall be limited in accordance with the TSP.

P.AUTHORIZED_USERS

Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.

P.I_AND_A

All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.

P.NEED_TO_KNOW

The TOE shall limit the access to information in protected resources to those authorized users who have a need to know that information.

P.ROLES

The users of the TOE shall use an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.2 Threats

T.AUDIT_COMPROMISE

A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker’s actions.

T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

T.SYSACC

A malicious process or user may gain unauthorized access to the authorized administrator account, or that of other trusted personnel.

T.TSF_COMPROMISE

A malicious user or process may cause configuration data (including cryptographic keys) to be inappropriately accessed (viewed, modified or deleted).

T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

T.UNDETECTED_ACTIONS

Failure of the TOE to detect and record attempts to perform unauthorized actions may occur.

T.UNIDENTIFIED_ACTIONS

An authorized administrator may not be able to read audit records stored in the audit trail

3.3 Assumptions

A.NO_EVIL

Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.

A.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on TOE servers, other than those services necessary for the operation, administration and support of the TOE.

A.PHYSICAL It is assumed that appropriate physical security is provided for both the TOE and calling applications within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

A. ENVIRONMENT It is assumed that the IT environment provides support commensurate with the expectations of the TOE.

4. Security Objectives

The following subsections describe objectives for the TOE and its environment that is consistent with the environment described in the previous section.

4.1 Security Objectives for the TOE

O.ACCESS
The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

O.ADMIN_ROLE
The TOE will provide authorized administrator roles to isolate administrative actions.

O.AUDIT_GENERATION
The TOE will provide the capability to detect and create records of security relevant events associated with users.

O.CRYPTOMODULE
The cryptomodule API in the IT environment generates cryptographic keys, performs cryptographic operations, and stores cryptographic keys (keys may be stored in either a FIPS 140-2 evaluated cryptomodule or in encrypted form in a database).

O.DISCRETIONARY_ACCESS
The TOE will control access to cryptographic keys based upon user identity, role, group membership, and access control lists.

O.USER_AUTHENTICATION
The TOE will verify the claimed identity of users.

O.USER_IDENTIFICATION
The TOE will uniquely identify users.

O.MANAGE
The TOE will provide all the functions and facilities necessary to support the authorized administrators and object owners in their management of the security of the TOE.

4.2 Security Objectives for the IT Environment

OE.AUDIT_REVIEW
The IT Environment will provide the ability to read from the audit trail.

OE.TIME
The IT environment will provide a time source that provides reliable time stamps.

OE.TOE_PROTECTION
The IT environment will provide protection to the TOE and its assets from external interference or tampering.

4.3 Security Objectives for the Non-IT Environment

OE.PERSON
Authorized administrators of the TOE shall be properly trained in the configuration and usage of the TOE and will follow the guidance provided. These users are not careless, negligent, or hostile.

OE.CONFIG
The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures by appropriately trained and trusted administrator personnel.

OE.INSTALL
The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.

OE.NO_GENERAL_PURPOSE
There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on TOE servers, other than those services necessary for the operation, administration and support of the TOE.

OE.PHYSICAL The environment in which the TOE operates is sufficient for secure operation. That the parts of the TOE critical to security policy are protected from physical attack and modification that might compromise the TOE security objectives.

OE.TRUST_IT Each IT entity the TOE relies on for security functions will be installed, configured, managed, maintained and provide the applicable security functions in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are satisfied by the TOE.

Table 1 – SFRs satisfied by the TOE

Requirement Class	Requirement Component
FAU: Security audit data generation	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
FCS: Cryptographic support	FCS_CKM.1: Cryptographic key generation
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1: Cryptographic operation
FDP: User Data Protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Subset attribute based access control
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MSA.1a: Management of security attributes
	FMT_MSA.1b: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_REV.1a: Revocation
	FMT_REV.1b: Revocation
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1: Basic internal TSF data transfer protection
FTA: TOE access	FTA_SSL.3: TSF-initiated termination

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

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 events:**

- **successful requests to perform an operation on an object covered by the SFP**
- **unsuccessful use of the authentication mechanism**
- **unsuccessful use of the user identification mechanism, including the user identity provided**
- **use of the management functions**
- **modifications to the group of users that are part of a role**
- **changes to the time**].

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, 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, [**object identity and operation**].

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**authorized administrator**] with the capability to read [**all audit information**] 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.

5.1.1.4 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.5 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform [***searches and sorting***] of audit data based on [**date&time, subject identity, event type**].

5.1.2 Cryptographic Support (FCS)

5.1.2.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**ANSI X9.31 RNG with 2-key Triple DES**] and specified cryptographic key sizes [**as listed in the table associated with FCS_COP.1**] that meet the following: [**Annex C to FIPS PUB 140-2**].

5.1.2.2 Cryptographic key destruction (FCS_CKM.4)

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**].

5.1.2.3 Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform [**encryption, decryption, and hashing**] in accordance with a specified cryptographic algorithm [**as listed in the following table**] and cryptographic key sizes [**also as listed in the following table**] that meet the following: [**standards, also as listed in the following table**].

Table 2 – Cryptographic key operations

Algorithm	Key size(s) - bits	Standard(s)
RSA Encryption	1024	PKCS-1, v 1.5

Algorithm	Key size(s) - bits	Standard(s)

AES	128, 192, 256	FIPS PUB 197
		Modes of operation:
		ECB(e/d; 128,192,256);
		CBC(e/d; 128,192,256);
		CFB8(e/d; 128,192,256);
		CFB128(e/d; 128,192,256);
		OFB(e/d; 128,192,256)

3DES	168	ANSI X9.52, FIPS PUB 46-3
		Modes of operation:
		TECB(e/d; KO 1,2,3);
		TCBC(e/d; KO 1,2,3);
		TCFB8(e/d; KO 1,2,3);
		TCFB64(e/d; KO 1,2,3);
		TOFB(e/d; KO 1,2,3)

MD5 Hash	N/A	RFC 1321
SHA-1 Hash	N/A	FIPS PUB 180-1

5.1.3 User Data Protection (FDP)

5.1.3.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] on [
a) subjects: all users;
b) objects: cryptographic keys ; and,
c) operations: all operations on the identified objects by subjects].

5.1.3.2 Subset attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to objects based on the following: [
a) subject attributes: user identity, assigned roles, group membership; and,
b) object attributes: owner, access control lists (ACLs)].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**if the user is assigned the required roles and one of the following conditions is met:**

- **if the user identity is equal to the object owner, the requested access is allowed; or**
- **if the ACL grants the requesting user identity the requested access, the requested access is allowed; or**
- **if the user identity is a member of a group and the ACL grants the group the requested access, the requested access is allowed**].

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

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**there are no explicit denial rules**].

5.1.4 Identification and Authentication (FIA)

5.1.4.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:[
a) user identity,
b) authentication data,
c) authentication method,
d) roles, and
e) group memberships].

5.1.4.2 User authentication before any action (FIA_UAU.2)

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.

5.1.4.3 User identification before any action (FIA_UID.2)

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

5.1.4.4 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**user identity, group membership and roles**].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**the subject identity, group memberships, and roles are assigned from TSF data that defines the applicable identified and authenticated user after a successful login**].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**subject security attributes cannot be changed**].

5.1.5 Security management (FMT)

5.1.5.1 Management of security attributes (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to restrict the ability to [***query, modify,* and *delete***] the security attributes [

- **user identity,**
- **user roles,**
- **group memberships**]

to [**the authorized administrator**].

5.1.5.2 Management of security attributes (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to restrict the ability to [***query, modify,* and *delete***] the security attributes [

- **access control lists (ACLs)**]

to [**the users who are assigned the Encryptor role and own a given object**].

5.1.5.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized administrators**] to specify alternative initial values to override the default values when an object or information is created.

5.1.5.4 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to [***query***] the [**audit records**] to [**authorized administrators**].

5.1.5.5 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to [***modify***] the [**user session timeout**] to [**authorized administrators**].

5.1.5.6 Revocation (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the [***subjects***] within the TSC to [**authorized administrators**].

FMT_REV.1a.2 The TSF shall enforce the rules [**the enforcement of subject attribute changes shall take effect before the next connection attempt**].

5.1.5.7 Revocation (FMT_REV.1b)

FMT_REV.1b.1 The TSF shall restrict the ability to revoke security attributes associated with the [***objects***] within the TSC to [**users who are assigned the Encryptor role and own a given object**].

FMT_REV.1b.2 The TSF shall enforce the rules [**the enforcement of object attribute changes shall take effect before the next access attempt related to that object**].

5.1.5.8 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [**manage users, manage roles, manage groups, manage audit logs, manage user session timeout, and manage ACLs of the cryptographic keys**].

5.1.5.9 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**Administrator role and Encryptor role**].

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

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Basic internal TSF data transfer protection (FPT_ITT.1)

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

5.1.7 TOE Access (FTA)

5.1.7.1 TSF-initiated termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [**period of user inactivity, as specified by an authorized administrator**].

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment of the TOE.

Table 3 – IT Environment Security Functional Components

Requirement Class	Requirement Component
FPT: Protection of the TSF	FPT_RVM.1: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation
	FPT_STM.1: Reliable time stamps

5.2.1 Protection of the TSF (FPT)

5.2.1.1 Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 The ~~TSF~~ **IT Environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.1.2 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~ **IT Environment** shall maintain a security domain for its own execution that protects it from interference and tampering by un-trusted subjects.

FPT_SEP.1.2 The ~~TSF~~ **IT Environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.1.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~ **IT Environment** shall be able to provide reliable time stamps for use by the TSF.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 , ALC_FLR.2, and AVA_MSU.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components. The following table identifies the SARs that are satisfied by the TOE:

Table 4 – EAL 2 and ALC_FLR.2, and AVA_MSU.1 assurance components

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.2: Configuration items
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.1: Descriptive high-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.1: Examination of guidance
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

5.3.1 Configuration Management (ACM)

5.3.1.1 Configuration Items (ACM_CAP.2)

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labeled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

ACM_CAP.2.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2d The developer shall use the delivery procedures.

ADO_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1d The developer shall provide a functional specification.

ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2c The functional specification shall be internally consistent.

ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4c The functional specification shall completely represent the TSF.

ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.
ADV_HLD.1.1C The presentation of the high-level design shall be informal.
ADV_HLD.1.2C The high-level design shall be internally consistent.
ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life cycle (ALC)

5.3.5.1 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall 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 documentation 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 corrected and the correction 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.

5.3.6 Tests (ATE)

5.3.6.1 Evidence of coverage (ATE_COV.1)

ATE_COV.1.1D The developer shall provide an analysis of the test coverage.

ATE_COV.1.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described 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.

5.3.6.2 Functional testing (ATE_FUN.1)

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, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 Independent testing - sample (ATE_IND.2)

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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Examination of guidance (AVA_MSU.1)

AVA_MSU.1.1d The developer shall provide guidance documentation.

AVA_MSU.1.1c The guidance documentation 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.

AVA_MSU.1.2c The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3c The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2e The evaluator shall repeat all configuration and installation procedures, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

5.3.7.2 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Developer vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1d The developer shall perform a vulnerability analysis.

AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security Audit

The TOE can generate audit events that are stored in a database in the IT Environment for the minimum level of audit, which includes:

- start-up and shutdown of the audit functions;
- successful requests to perform an operation on an object covered by the SFP;
- unsuccessful use of the authentication mechanism;
- unsuccessful use of the user identification mechanism, including the user identity provided;
- use of the management functions;
- modifications to the group of users that are part of a role; and,
- changes to the time.

Audit records include date and time of the event, user ID that caused the event to be generated, unique ID of the Tricryption KeyServer, and event specific data (e.g., the target object and operation being performed). The IT environment is relied on to provide a reliable time stamp, as well as to protect the audit trail as part of the TOE security domain from unauthorized access.

The TOE provides an interface that allows authorized administrator to review all of the data in the audit trail.

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

- FAU_GEN.1: The TOE generates audit events identified above. Access control and management related events are audited.
- FAU_GEN.2: Audit records generated associate the user ID with the event.
- FAU_SAR.1, FAU_SAR.3: The TOE provides a LogReporter interface to review the audit records. Audit records can be searched and sorted based on date & time, user identity, and event type.

6.1.2 Cryptographic Protection

The TOE includes a FIPS 140-2 certified cryptographic module (certificate #1094), the Tricryption Cryptographic Module (or Tricryption Cryptomodule). This module is responsible to create and destroy keys and also to perform other cryptographic operations (e.g., encryption and decryption) in accordance with a wide range of algorithm specifications (see FCS_COP.1, above). More specific information about this module, including its abilityies to create and destroy keys, can be found here: http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1094.pdf.

The TOE utilizes the cryptographic support provided by the cryptographic module in order to encrypt and decrypt the communication channels between the TOE components, should they be distributed, in order to protect that data from unauthorized modification or disclosure.

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

- FCS_CKM.1: The TOE includes a cryptographic module capable of generating keys suitable for the algorithms specified in FCS_COP.1 in accordance with FIPS 140-2.
- FCS_CKM.4: The TOE includes a cryptographic module capable of destroying keys via zeroization in accordance with FIPS 140-2.
- FCS_COP.1 The TOE includes a cryptographic module capable of implementing all of the algorithms identified in FCS_COP.1.
- FPT_ITT.1: The TOE encrypts communication channels between distributed TOE components.

6.1.3 User Data Protection

The TOE implements a Discretionary Access Control (DAC) and role-based policies for object access based on:

- user identity,
- group membership,
- cryptographic key identity,
- access control list (ACL), and
- assigned roles (permission).

The TOE objects that are subject to this policy are encrypted keys that are used to protect the confidentiality of user data from applications outside of the TOE boundary. Key identifiers (Key IDs) are generated by the TOE and assigned to keys that are generated and encrypted/decrypted by the cryptomodule in the IT environment with the call of the cryptomodule API in the Tricryption KeyServer. The Key IDs are then encrypted by the cryptomodule in the IT environment with the call of the cryptomodule API in the Tricryption KeyServer; encrypted Key IDs are also known as Hidden Links.

The TOE restricts access to encrypted keys using ACLs. ACLs are used to grant access to encrypted keys to individual users. Users who are identified as having been granted access to an encrypted key may access the encrypted key. Users who are identified as members of groups that have been granted access to an encrypted key may access the encrypted key. Encrypted keys have owners, and owners may grant permission to access encrypted keys that they own to other users by updating corresponding ACLs. Note that when a user is added to an ACL, an additional key is not generated and the protected data is not re-encrypted using the new user's key, i.e. the TOE is not using a public key cryptography scheme to protect data.

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

- FDP_ACC.1: All Tricryption KeyServer subjects are subject to the DAC SFP for all available operations on encrypted keys.

- FDP_ACF.1: The TOE is able to restrict access to encrypted keys using ACLs. ACLs are used to grant access to individual Tricryption KeyServer users.
- FMT_MSA.1b: The TOE restricts the ability to query, modify, and delete access control lists (ACLs). to an user by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object and users in the encryptor role. When the server receives a request to access the ACL, the server will perform a check to determine if the caller is the owner of the object or if the caller has the encryptor role.
- FMT_MSA.3: By default every key is created with the creator as the owner. Subsequently, access can be granted to other users.
- FMT_REV.1b: Users can revoke security attributes associated with the cryptographic keys by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object and users in the encryptor role. For any attempts to revoke security attributes associated with cryptographic keys, the TSF will perform an access check to determine if the caller is the owner of the object or if the caller has the encryptor role.
- FMT_SMF.1: Administrators are able to perform all management functions, including: managing users (including TOE IT entities) and managing encrypted keys.

6.1.4 Identification and Authentication

The TOE defines users in terms of:

- user identity,
- authentication data,
- authentication method,
- roles, and
- group memberships.

The TOE provides its own username and password authentication mechanism utilizing Secure Remote Password (SRP). This is impolemented by incorporating open source libraries (http://srp.stanford.edu/).

Each user is identified and authenticated when a connection is made to the TOE and the TOE instantiates a task with the user attributes for the duration of the connection. When the task is created, it is assigned the user's identity, groups, and role per the user definition. Once assigned the attributes cannot be changed.

The TOE provides Tricryption Agent API interfaces that is used by non-administrative users to access TOE services. The TOE provides Tricryption administrator console component GUI interfaces that can be used by administrative users to manage TOE security functions. Both non-administrative users and administrative users are required to provide username and password in order to be authenticated using either the TOE username/password mechanism.

Every user is assigned one or more roles and each role is itself assigned one or more operations, commonly referred to as permissions. As a result, each user is assigned specific permissions, which govern the user's activities in relation to the TOE. The operations that may be assigned to various roles are preset. By default, most of the operations are initially assigned to the preset Administrator role. Additional roles and usernames may be added, updated and removed by the administrator.

The TOE implements password composition requirements and minimum password lengths. Passwords must contain at least 8 characters of mixed alphanumerics. There must be at least one change of case and one or more digits. When a user is assigned a password, the password will be rejected unless it satisfies these minima requirements.

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

- FIA_ATD.1: For each user account, the TOE keeps at least the following information: user identity, authentication data, authentication method, roles, and, group memberships.

- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.
- FIA_USB.1: The TOE instantiates users as tasks with an assigned identity, groups, and roles. A given user has only a defined attributes and is unable to change those attributes once logged in.

6.1.5 Security Management

The TOE allows the definition of user groups, there are no pre-defined groups.

To facilitate data sharing, multiple users may be joined together to form groups. Any privileges granted to a group are applied to all current membership. Effective access control may involve adding and removing user groups and managing group memberships. Adding a new group to the Tricryption KeyServer involves two steps:

- Adding a new group and
- Assigning users to that group.

Removing a member of a group revokes all privileges granted to the member by the group. An existing group name cannot be changed.

The TOE defines the following roles:

- Encryptor and
- Administrator.

Note that members of the pre-defined role "Encryptor" are simply considered users.

An authorized administrator is a user that has been assigned the administrator role. Users are all other users, including users assigned the Encryptor role.

The TOE defines the following operations for the Encryptor role:

- Encrypt data ,
- Decrypt data,
- Encrypt file,
- Decrypt file,
- Get an ACL, and
- Update an ACL including adding, removing and updating the ACL entries.

The TOE defines the following operations for the Administrator role:

- Add a principal (user),
- Add a group,
- Remove a principal (user),
- Remove a group,
- Update a principal's information,
- Update a group's information,
- Search for one or more principals,
- Get a list of all principals,
- Add a principal to a group,
- Remove a principal from a group,

- Update a principal’s password,
- Create a role,
- Remove a role,
- Get one or more roles,
- Update a role including adding and removing operation,
- Get a list of all operations,
- Get configuration, and
- Update configuration.

Groups and Roles differ in that Groups are a collection of multiple users with common privileges, whereas a Role is an allowable set of operations assigned to one or more users. The preset Administrator role is assigned by default to the Tricryption KeyServer Administrator specified during product installation, and is the authorized administrator. The other preset role, Encryptor is initially unassigned, and is assigned to specific users by the authorized administrator. A minimum of one user, the authorized administrator, is necessary for the Tricryption KeyServer to function.

The Tricryption administrator console component (Tricryption Manager) provides a GUI interface that is accessible using Java application interfaces (the Tricryption Manager runs as a Java application in the context of the JVM in the IT environment). The administrator console interfaces can be used by administrators to perform the following:

- manage users, groups, roles and
- manage TOE configuration.

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

- FMT_MSA.1a: The TOE restricts the ability to query, modify, and delete user identity, roles, permissions, to an administrator by restricting access to administrator console interfaces and to the API.
- FAU_SAR.2: Only the authorized administrator can query the audit trail.
- FMT_MTD.1a: Only the authorized administrator can query the audit trail.
- FMT_MTD.1b: Only the administrator role can modify user session timeout values by restricting access to administrator console interfaces.
- FMT_REV.1a: Only the administrator role can revoke security attributes associated with the users by restricting access to administrator console interfaces and to the API.
- FMT_SMF.1: Administrators are able to perform all management functions, including: managing users (including TOE IT entities). Owner of the cryptographic key is able to manage the ACLs of the cryptographic keys.
- FMT_SMR.1: An authrorized administrator is a user that has been assigned the Administrator role. Users are all other users, including users assigned the Encryptor role.

6.1.6 TOE Access

The TOE can terminate Tricryption administrator console component sessions after an administrator-configured (using the Tricryption administrator console) period of time.

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

- FTA_SSL.3: The TOE terminates an interactive user session after a period of inactivity.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by ERUCES ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. ERUCES performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, and the CM documentation.

These activities are documented in:

- ERUCES *Tricryption KeyServer and Agent Version 6.2 Configuration Management Plan*

The Configuration management assurance measure satisfies the following EAL 2 assurance requirements:

- ACM_CAP.2: The TOE was developed using a CM system, and is labeled with a unique reference.

6.2.2 Delivery and operation

ERUCES provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. ERUCES' delivery procedures describe all applicable procedures to be used to prevent in appropriate access to the TOE. ERUCES also provides documentation that describes the steps necessary to install the TOE in accordance with the evaluated configuration.

These activities are documented in:

- ERUCES *Tricryption KeyServer and Agent version 6.2 Delivery Procedures*
- Tricryption Suite Installation Guide, Version 6.2

The Delivery and operation assurance measure satisfies the following EAL 2 assurance requirements:

- ADO_DEL.1: ERUCES has documented and implemented their delivery procedures.
- ADO_IGS.1: ERUCES has documented and supplies start-up documentation for the secure installation of the TOE.

6.2.3 Development

ERUCES documents the design of the TOE. These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

These activities are documented in:

- Tricryption Suite Architecture, Use Case Model Top Level Packages
- ERUCES *Tricryption KeyServer and Agent 6.2 High-Level Design*
- ERUCES *Tricryption KeyServer and Agent 6.2 Functional Specification*

The Development assurance measure satisfies the following EAL 2 assurance requirements:

- ADV_FSP.1: The TOE's Functional Specification describes the TSF and external interfaces.
- ADV_HLD.1: The TOE's High-level Design document describes the TSFs in terms of subsystems.
- ADV_RCR.1: The TOE's Correspondence document accounts for all adjacent pairs of TSF representations.

6.2.4 Guidance documents

ERUCES provides administrator and guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE. A separate document is available for software developers to implement the TOE into integrated applications.

These activities are documented in:

- Tricryption KeyServer Administration Guide, Version 6.2
- Tricryption Sutie Installation Guide Version 6.2
- Tricryption Software Development Kit 6.2

The Guidance documents assurance measure satisfies the following EAL 2 assurance requirements:

- AGD_ADM.1: The TOE's administrative guidance document describes how to administer the TOE in a secure manner.
- AGD_USR.1: The TOE's administrative guidance document describes the functions and interfaces available to the non-administrative users of the TOE.

6.2.5 Life cycle support

ERUCES employs a process where security flaws discovered by customers and ERUCES are tracked and corrected by the developer. ERUCES bug reconciliation process provides assurance that the TOE is maintained and flaws are corrected in the TOE. To meet ALC_FLR.2, ERUCES has a flaw remediation process that tracks reported security flaws in each release of the TOE. The flaw procedure accepts and acts upon reports of security flaws and requests for corrections to those flaws by outside sources, ERUCES developers and customers, and issues remediation guidance to TOE clients.

These activities are documented in:

- ERUCES, Inc. QA Procedural Standards

The Life cycle support assurance measure satisfies the following assurance requirements:

- ALC_FLR.2: ERUCES has a flaw remediation procedure, with guidance for reporting security flaws supplied to the TOE users.

6.2.6 Tests

The test documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

These activities are documented in:

- ERUCES Tricryption KeyServer and Agent Version 6.2 Test Document (COV and FUN)

The Tests assurance measure satisfies the following EAL 2 assurance requirements:

- ATE_COV.1: The TOE's Test Coverage Report demonstrates the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
- ATE_FUN.1: The TOE's Function Test Report demonstrates that each tested security function behaved as specified.
- ATE_IND.2: The independent evaluation of the TOE verifies the developer's test results.

6.2.7 Vulnerability assessment

ERUCES has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-basic.

To meet AVA_MSU.1, ERUCES appears to have guidance documentation that identifies all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

ERUCES performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE.

These activities are documented in:

- ERUCES Tricryption KeyServer and Tricryption Agent Vulnerability Assessment
- Tricryption KeyServer Administration Guide, Version 6.2

The Vulnerability assessment assurance measure satisfies the following EAL 2 assurance requirements:

- AVA_SOF.1: The developer's analysis for each mechanism identified in the ST as having a strength of TOE security function claim shows that it meets or exceeds the minimum strength level defined in the ST.
- AVA_VLA.1: The developer's analysis shows, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. 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;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of threats and usage assumptions by the security objectives.

Table 5 – Environment to Objective Correspondence

	P.ACCOUNTABILITY	P.AUTHORIZATION	P.AUTHORIZED_USERS	P.I_AND_A	P.NEED_TO_KNOW	P.ROLES	T.AUDIT_COMPROMISE	T.MASQUERADE	T.SYSACC	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNIDENTIFIED_ACTIONS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A. ENVIRONMENT
O.ACCESS		X	X		X				X		X						
O.ADMIN_ROLE	X					X											
O.AUDIT_GENERATION	X											X					
OE.AUDIT_REVIEW												X					
O.DISCRETIONARY_ACCESS					X						X						
O.MANAGE									X				X				
O.USER_AUTHENTICATION				X				X	X								
O.USER_IDENTIFICATION	X	X		X	X			X	X								
OE.CRYPTOMODULE										X	X						
OE.TIME	X											X					
OE.TOE_PROTECTION										X							
OE.PERSON									X				X				
OE.CONFIG														X			
OE.INSTALL														X			

	P.ACCOUNTABILITY	P.AUTHORIZATION	P.AUTHORIZED_USERS	P.I_AND_A	P.NEED_TO_KNOW	P.ROLES	T.AUDIT_COMPROMISE	T.MASQUERADE	T.SYSACC	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNIDENTIFIED_ACTIONS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A. ENVIRONMENT
OE.NO_GENERAL_PURPOSE															X		
OE.PHYSICAL							X		X	X	X	X				X	
OE.TRUST_IT																	X

8.1.1.1 P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their actions within the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.ADMIN_ROLE: The TOE will provide authorized administrator roles to isolate administrative actions.
- O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security relevant events associated with users.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.
- OE.TIME: The IT environment will provide a time source that provides reliable time stamps for audit records.

8.1.1.2 P.AUTHORIZATION

The abilities of users of the TOE shall be limited in accordance with the TSP.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.3 P.AUTHORIZED_USERS

Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

8.1.1.4 P.I_AND_A

All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.

This Organizational Policy is satisfied by ensuring that:

- O.USER_AUTHENTICATION: The TOE will verify the claimed identity of users.

- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.5 P.NEED_TO_KNOW

The TOE shall limit the access to information in protected resources to those authorized users who have a need to know that information.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.DISCRETIONARY_ACCESS: The TOE will control access to cryptographic keys based upon user identity, role, group membership, and access control lists.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.6 P.ROLES

The users of the TOE shall use an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

This Organizational Policy is satisfied by ensuring that:

- O.ADMIN_ROLE: The TOE will provide authorized administrator roles to isolate administrative actions.

8.1.1.7 T.AUDIT_COMPROMISE

A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.

This Threat is satisfied by ensuring that:

- OE.PHYSICAL: The environment must address the possible compromise of audit data due to physical means.
- OE.TOE_PROTECTION: The IT environment must also protect itself and its assets.

8.1.1.8 T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

This Threat is satisfied by ensuring that:

- O.USER_AUTHENTICATION: The TOE will verify the claimed identity of users.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.9 T.SYSACC

A malicious process or user may gain unauthorized access to the authorized administrator account, or that of other trusted personnel.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.MANAGE: The TOE will provide all the functions and facilities necessary to support the authorized administrators and object owners in their management of the security of the TOE.
- O.USER_AUTHENTICATION: The TOE will verify the claimed identity of users.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

- OE.PERSON: The TOE guidance includes complete and clear administration guidance.
- OE.PHYSICAL: The environment must address the possible unauthorized access to administrative accounts due to physical means.

8.1.1.10 T.TSF_COMPROMISE

A malicious user or process may cause configuration data (including cryptographic keys) to be inappropriately accessed (viewed, modified or deleted).

This Threat is satisfied by ensuring that:

- OE.CRYPTOMODULE: The IT environment generates cryptographic keys, performs cryptographic operations, and stores cryptographic keys (keys may be stored in either a FIPS 140-2 evaluated cryptomodule or in encrypted form in a database).
- OE.TOE_PROTECTION: The IT environment will provide protection to the TOE and its assets from external interference or tampering.

8.1.1.11 T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.DISCRETIONARY_ACCESS: The TOE will control access to cryptographic keys based upon user identity, role, group membership, and access control lists.
- OE.CRYPTOMODULE: The IT environment generates cryptographic keys, performs cryptographic operations, and stores cryptographic keys (keys may be stored in either a FIPS 140-2 evaluated cryptomodule or in encrypted form in a database).
- OE.PHYSICAL: The environment must address the possible unauthorized access to administrative accounts due to physical means.
- OE.TOE_PROTECTION: The IT environment will provide protection to the TOE and its assets from external interference or tampering.

8.1.1.12 T.UNDETECTED_ACTIONS

Failure of the TOE to detect and record attempts to perform unauthorized actions may occur.

This Threat is satisfied by ensuring that:

- O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security relevant events associated with users.
- OE.AUDIT_REVIEW: The IT Environment will provide the ability to read from the audit trail.
- OE.TIME: The IT environment will provide a time source that provides reliable time stamps for use in audit records.
- OE.PHYSICAL: The environment must address the possible unauthorized access to administrative accounts due to physical means.

8.1.1.13 T.UNIDENTIFIED_ACTIONS

An authorized administrator may not be able to read audit records stored in the audit trail.

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE will provide all the functions and facilities necessary to support the authorized administrators and object owners in their management of the security of the TOE.
- OE.PERSON: The TOE guidance includes complete and clear administration guidance.

8.1.1.14 A.NO_EVIL

Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.

This Assumption is satisfied by ensuring that:

- OE.INSTALL: The TOE guidance includes the necessary installation instructions that detail how to securely install the TOE.
- OE.CONFIG: Authorized administrators are trained and trusted to properly configure the IT environment so it enforces its security policies.

8.1.1.15 A.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on TOE servers, other than those services necessary for the operation, administration and support of the TOE.

This Assumption is satisfied by ensuring that:

- OE.NO_GENERAL_PURPOSE: The TOE server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.

8.1.1.16 A.PHYSICAL

It is assumed that appropriate physical security is provided for both the TOE and calling applications within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: The environment must address the possible unauthorized access to administrative accounts due to physical means.

8.1.1.17 A. ENVIRONMENT

It is assumed that the IT environment provides support commensurate with the expectations of the TOE.

This Assumption is satisfied by ensuring that:

- OE.TRUST_IT: The IT entities in the environment are correctly installed, configured, managed, maintained and provide the applicable security functions.

8.2 Security Functional Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that the following table indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

Table 6 – Objective to Requirement Correspondence

	O.ACCESS	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.CRYPTOMODULE	O.DISCRETIONARY_ACCESS	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	O.MANAGE	OE.AUDIT_REVIEW	OE.TIME	OE.TOE_PROTECTION
FAU_GEN.1			X								
FAU_GEN.2			X								
FAU_SAR.1									X		
FAU_SAR.2									X		
FAU_SAR.3									X		
FCS_CKM.1				X							
FCS_CKM.4				X							
FCS_COP.1				X							
FDP_ACC.1	X				X						
FDP_ACF.1	X				X						
FIA_ATD.1							X				
FIA_UAU.2						X					
FIA_UID.2							X				
FIA_USB.1							X				
FMT_MSA.1a					X			X			
FMT_MSA.1b					X			X			
FMT_MSA.3								X			
FMT_MTD.1a								X			
FMT_MTD.1b								X			
FMT_REV.1a	X							X			
FMT_REV.1b	X							X			
FMT_SMF.1					X			X			
FMT_SMR.1		X						X			
FPT_RVM.1											X
FPT_SEP.1											X
FPT_STM.1			X							X	
FPT_ITT.1	X										
FTA_SSL.3	X										

8.2.1.1 O.ACCESS

The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: All Tricryption KeyServer subjects are subject to the DAC SFP for all available operations on encrypted keys.
- FDP_ACF.1: The TOE is able to restrict access to encrypted keys using ACLs. ACLs are used to grant access to individual Tricryption KeyServer users.
- FMT_REV.1a: Only the administrator role can revoke security attributes associated with the users by restricting access to administrator console interfaces.
- FMT_REV.1b: Users can revoke security attributes associated with the cryptographic keys by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object.
- FPT_ITT.1: The TOE protects communication between its components from modification and disclosure using cryptographic mechanisms.
- FTA_SSL.3: The TOE terminates an interactive user session after a period of inactivity.

8.2.1.2 O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: An authrorized administrator is a user that has been assigned administrator role. Users are all other users, including users assigned the Encryptor role.

8.2.1.3 O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE generates audit events a minimum level of audit. Access control and management events are audited.
- FAU_GEN.2: Audit records generated associate the user ID with the event.
- FPT_STM.1: The IT environment is required to provide reliable time stamps.

8.2.1.4 O.CRYPTOMODULE

The IT environment generates cryptographic keys, performs cryptographic operations, and stores cryptographic keys (keys may be stored in either a FIPS 140-2 evaluated cryptomodule or in encrypted form in a database).

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to generate cryptographic keys for other cryptographic operations.
- FCS_CKM.4: The TOE is required to destroy cryptographic keys approrpaitely.
- FCS_COP.1: The TOE is required to perform cryptographic operations.

8.2.1.5 O.DISCRETIONARY_ACCESS

The TOE will control access to cryptographic keys based upon user identity, role, group membership, and access control lists.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: All Tricryption KeyServer subjects are subject to the DAC SFP for all available operations on encrypted keys.
- FDP_ACF.1: The TOE is able to restrict access to encrypted keys using ACLs. ACLs are used to grant access to individual Tricryption KeyServer users.
- FMT_MSA.1a: The TOE restricts the ability to query, modify, and delete user identity, roles, permissions, to an administrator by restricting access to administrator console interfaces.
- FMT_MSA.1b: The TOE restricts the ability to query, modify, and delete access control lists (ACLs).to an user by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object.
- FMT_SMF.1: Administrators are able to perform all management functions, including: managing users (including TOE IT entities). Owner of the cryptographic key is able to manage the ACLs of the cryptographic keys.

8.2.1.6 O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.

8.2.1.7 O.USER_IDENTIFICATION

The TOE will uniquely identify users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1: For each user account, the TOE keeps at least the following information: user identity, authentication data, authentication method, roles, and, group memberships.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user.

8.2.1.8 O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators and object owners in their management of the security of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MSA.1a: The TOE restricts the ability to query, modify, and delete user identity, roles, permissions, to an administrator by restricting access to administrator console interfaces and API.
- FMT_MSA.1b: The TOE restricts the ability to query, modify, and delete access control lists (ACLs).to an user by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object.
- FMT_MSA.3: By default every key is created with the creator as the owner. Subsequently, access can be granted to other users.
- FAU_SAR.1: Only an authorized administrator can query the audit trail.

- FMT_MTD.1a: Only an authorized administrator can query the audit trail by restricting access to administrator console interfaces.
- FMT_MTD.1b: Only the administrator role can modify user session timeout values by restricting access to administrator console interfaces.
- FMT_REV.1a: Only the administrator role can revoke security attributes associated with the users by restricting access to administrator console interfaces and API.
- FMT_REV.1b: Users can revoke security attributes associated with the cryptographic keys by the server restricting access to these functions that are accessible using the client API to authenticated users that own a given object.
- FMT_SMF.1: Administrators are able to perform all management functions, including: managing users (including TOE IT entities). Owner of the cryptographic key is able to manage the ACLs of the cryptographic keys.
- FMT_SMR.1: An authorized administrator is a user that has been the administrator role. Users are all other users, including users assigned the Encryptor role.

8.2.1.9 OE.AUDIT_REVIEW

The TOE will provide the ability to read from the audit trail.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1, FAU_SAR.2, and FAU_SAR.3: The TOE provides (only) the authorized administrator with the capability to read all audit information from the audit records. Note that the Tricryption LogReporter provides GUI interfaces for reviewing (and searching) the audit records.

8.2.1.10 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment is required to provide reliable time stamps.

8.2.1.11 OE. TOE_PROTECTION

The IT environment will provide protection to the TOE and its assets from external interference or tampering.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_RVM.1: The IT environment is required to grant access to protected objects only after it makes informed access decisions.
- FPT_SEP.1: The IT environment is required to protect itself and separate the contexts of its users.

8.3 Security Assurance Requirements Rationale

The base assurance level of EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1 because flaw remediation procedures provide greater assurance that security-related bugs will be fixed in a widely distributed commercial product, and misuse investigates whether the TOE can be configured or used in a manner that is insecure (but that an administrator or user would reasonably believe to be secure).

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 a relatively benign environment with good physical access security and competent administrators. As such, EAL 2+ is appropriate to provide the assurance necessary to counter this potential for attack.

8.4 Strength of Functions Rationale

The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a low attack potential. As such, the minimum strength of function level, together with the SOF-basic claim, is appropriate for the intended environment and consistent with the security objectives for the TOE. Note that the only applicable mechanisms (i.e., those that are probabilistic or permutational) are related to authentication (i.e., FIA_UAU.2, and FIA_UID.2). The overall strength of function claim of basic is believed to be commensurate with the overall assurance claim of EAL 2 augmented with ALC_FLR.2 and AVA_MSU.1.

8.5 Requirement Dependency Rationale

The following table demonstrates that all dependencies among the claimed security requirements are satisfied and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

Table 7 – Dependency Analysis

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.2
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_COP.1 and 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 FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	FCS_CKM.1 and FCS_CKM.4 FMT_MSA.2 is **not** included; however the TOE includes a FIPS-evaluated cryptographic engine to perform all cryptographic operations. Dependencies will have been satisfied as part of obtaining FIPS certification.
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1 and FMT_MSA.3
FIA_ATD.1	none	none
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	none	none
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1a	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.1
FMT_MSA.1b	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.1
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_REV.1a	FMT_SMR.1	FMT_SMR.1
FMT_REV.1b	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_RVM.1	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none
FPT_ITT.1	none	none

ST Requirement	CC Dependencies	ST Dependencies
FTA_SSL.3	none	none
ACM_CAP.2	none	none
ADO_DEL.1	none	none
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1	ADV_RCR.1
ADV_HLD.1	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.1 and ADV_RCR.1
ADV_RCR.1	none	none
AGD_ADM.1	ADV_FSP.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.1
ALC_FLR.2	none	none
ATE_COV.1	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.1 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.1 and ADV_HLD.1
AVA_VLA.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1

Functional component FCS_COP.1 depends on the following functional component FMT_MSA.2 Secure Security Attributes. Given that the cryptographic modules is FIPS PUB 140-2 compliant, secure key values will have been ensured via that certification. For more information, refer to FIPS PUB 140-2.

8.6 Explicitly Stated Requirements Rationale

There are no explicitly stated requirements in this ST.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. The following table demonstrates the relationship between security requirements and security functions.

Table 8 – Security Functions vs. Requirements Mapping

	Security audit	Cryptographic Protection	User data protection	Identification and authentication	Security management	TOE access
FAU_GEN.1	X					
FAU_GEN.2	X					
FAU_SAR.1	X					
FAU_SAR.2					X	
FAU_SAR.3	X					
FCS_CKM.1		X				
FCS_CKM.4		X				
FCS_COP.1		X				
FDP_ACC.1			X			
FDP_ACF.1			X			
FIA_ATD.1				X		
FIA_UAU.2				X		
FIA_UID.2				X		
FIA_USB.1				X		
FMT_MSA.1a					X	
FMT_MSA.1b			X			
FMT_MSA.3			X			
FMT_MTD.1a					X	
FMT_MTD.1b					X	
FMT_REV.1a					X	
FMT_REV.1b			X			
FMT_SMF.1			X		X	
FMT_SMR.1					X	
FPT_ITT.1		X				
FTA_SSL.3						X

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.