Security Target

KeyOne 3.0

KeyOne 3.0 Security Target

Safelayer Secure Communications, S.A.

Phone: +34 93 508 80 90

Fax: +34 93 508 80 91

Web: www.safelayer.com

Email: support@safelayer.com

CONTENTS

CHAPTER 1

1 Introduction

1.1 Identification

Document ID	**B4E6DBC0 v1.39**
Title	**Security Target KeyOne 3.0**
Issue Date	**October 24, 2006**
Release ID	**3.0 04S2R1**
Authors	**Safelayer Secure Communications S.A..**
State	**Issued**
CC Version	**2.2**
Evaluated TOE	**KeyOne 3.0 04S2R1: KeyOne CA, KeyOne LRA, KeyOne RA, KeyOne VA and KeyOne TSA** **Patches: 3.0_04S2R1_B01, 3.0_04S2R1_B02, 3.0_04S2R1_B03, 3.0_04S2R1_B04, 3.0_04S2R1_B05, 3.0_04S2R1_B06, 3.0_04S2R1_B07, 3.0_04S2R1_B08**

In order to fulfill with the EAL4+ security guarantees of the KeyOne product included in this Security Target, the license file used in the TOE does not have to allow the execution of scripts launched in unsecure mode (activation of the `--unsecure` flag). For more information about the license file, see the Appendix A Considerations about the license file, page 235.

1.2 Overview

The purpose of this ST is to specify functional and assurance security requirements implemented by KeyOne 3.0 04S2R1 TWS, which is the Target of Evaluation.

The content of the document is organized in the following chapters:

Chapter 1, provides labelling and descriptive information about the ST and the TOE that it refers to, a TOE summarize in narrative form and a conformance claim with CC requirements.

Chapter 2, provides a description of TOE services, gives an overview of the TOE users who will interact with it, describes the layout physical and logical architectures of the system and the contribution of each subsystem to the identified services. Finally, a list of the most common security services covered by the TOE and potential business applications where it should be useful.

Chapter 3, provides a security problem definition, showing the secure usage assumptions, assets, threats, and organizational security policies that must be upheld, protected, countered and enforced by the TOE and its operational environment.

Chapter 4, contains the solution to this security problem by providing security objectives for the TOE and for the environment.

Chapter 5, provides a translation of the security objectives into a set of functional and assurance requirements in the form of CC part 2 requirements, extended functional requirements, CC part 3 requirements and extended assurance requirements.

Chapter 6, provides an explanation of how these security requirements are implemented in the TOE.

Chapter 7, contains conformance claims with Protection Profile and international standards.

Chapter 8, provides security objectives rationale showing that the security problem is solved if all the security objectives are achieved, and security requirements rationale demonstrating the effective tracing between them and the security objectives. Two more sections are included to demonstrated the consistency and mutually supportive of the whole.

Chapter 9, states the security policy enforced by this security target

Chapter 10, includes a glossary of terms used inside this document, a bibliography applicable to the scope of the ST, and a list of Acronyms.

The TOE defined by this ST provides the following CSP services:

- Registration of subscriber information (Registration Service)
- Certificate generation (Certificate Generation Service)
- Certificate revocation management (Revocation Management Service)
- Certificate revocation status provision (Revocation Status Service)
- Time Stamping Functions (Time Stamping Service)
- Signature-Creation/Secure-Signature-Creation Device production (Subscriber device provision service)

1.3 Conformance

KeyOne 3.0 conforms to the "Certificate Issuing and Management Components Family of Protection Profiles" (CIMC) Security Level 3 Protection Profile, version 1.0, October 31, 2001, National Security Agency (NSA). Additionally KeyOne 3.0 conforms

to all the Assurance Requirements for the EAL4 Common Criteria certification level, augmented with ALC_FLR.2.

In the construction of the security requirements the following inputs has been used:

a. Security functional requirements from part 2 of CC (Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.2, January 2004).

b. Security assurance requirements from part 3 of CC extracted from package EAL4 (Common Criteria for Information Technology Security Evaluation Part 2: Security assurance requirements, Version 2.2, January 2004).

c. Extended security functional requirements, expressed in the format of CC, from "Certificate Issuing and Management Components Family of Protection Profiles" (CIMC) Security Level 3 Protection Profile, version 1.0, October 31, 2001, National Security Agency (NSA).

1.4 Conventions

The objective of this section is to aid the reader by the inclussion of specific style and clarifying information conventions.

- Whenever an operation has been applied to a security functional requirement, the type of operation will be indicated, and the corresponding text will appear in italics. Both the type of operation and the corresponding text will be included in brackets (e.g., [selection : *event type*], [assignment : *no additional attributes*]).
- Whenever a security functional requirement has been used more than once this document, the title of the security functional requirement is followed by an iteration number (e.g., iteration 1) to distinguish between the different iterations of the security functional requirement.
- The instantiated operations from the CIMC Protection Profile will appear in italics in brackets (e.g., The [*IT environment*] shall provide the audit records in a manner suitable for the user to interpret the information).

CHAPTER 2

2 TOE Description

The TOE included in this Security Target has been designed and implemented in order to accomplish with the following Directives, Recommendations and Requirements:

- European Community. *Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a community framework for electronic signatures*, 1999.
- CEN/ISSS Workshop on Electronic Signatures. *Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures*, June 2003.
- ETSI TS 101 456, *Policy Requirements for Certification Authorities Issuing Qualified Certificates*.

Nevertheless the target of this evaluation it is not check the conformance of this product against the previously mentioned security specifications[1].

Although [Eur99b] has a very general approach and speaks of electronic signatures of any kind, the underlying assumption in this document is that electronic signatures are created by means of public key cryptography, that the subscriber uses a cryptographic key pair consisting of a private and public component, and that a certificate produced by a system considered in this document essentially binds the public key of the subscriber to the identity and possibly other information of the subscriber by means of an electronic signature which is created with the private key (certificate signing key) of the TOE. Other forms of electronic signatures are outside the scope of this document.

Although security requirements for the optional Subscriber Device Provision Service, which provides SCD/SSCD provision to Subscribers is included within the scope of this ST, requirements of the actual SSCD devices themselves, as used by Subscribers of the TOE, are outside the scope of this document. Security requirements for SSCDs are provided in the separate document Secure Signature Creation Devices [CEN01b].

Following the principles of [Eur99b] this ST is as technology neutral as possible. It does not require or define any particular communication protocol or format for electronic signatures, certificates, certificate revocation lists, certificate status information and time stamps, but those international standards which ensure global interoperability. It only assumes certain types of information to be present in the certificates in accordance with Annex I of the European Directive. Interoperability between TOE systems and subscriber systems is outside the scope of this document

[1] *This statement is applicable to any reference to the above directives and security guidelines contained in this Security Target.*

2.1 Description of the Trustworthy System KeyOne 3.0

The KTS, within this specification, provides and manages certificates used for the support of electronic signatures. It is a primary assumption that the TOE will use a Public Key Infrastructure (PKI) for the management of certificates. The adopted approach of this specification is for a TOE to offer a number of services, each service having defined functions to facilitate service delivery. Each defined function is required to meet minimum-security standards thus achieving trustworthy status.

The TOE consists of a number of subsystems each providing specific TOE functionality. Although this specification considers security requirements of the subsystems involved in the TOE's service, the aim is to provide the Subscriber and Relying Party a single view of the TOE and hence a single view of the subsystems employed by it. To ensure this, the customer interface, in this specification, is to the "TOE Service" and not directly to the individual services offered by the TOE. As subsystems are further decomposed, any functionality, defined by other acceptable standards has been referenced.

The TOE provides services by deploying subsystems with Core Functionality and provides optional services by deploying subsystems with Supplementary Functionality. The TOE implements the Core Functionality to meet some requirements of [CEN01c]. The TOE also implements some optional services, in addition to some mandatory services, as defined by [CEN01c], and the security requirements specified in [CEN01c] for that service.

In effect the TOE deploys subsystems meeting General and Core Security Requirements. It is important to note that this technical/security integration does not necessarily impede on the freedom of the TOE to run the different components of the service using different business entities.

2.1.1 TOE Core Services

The core services the TOE provides are:

Registration Service: Verifies the identity and, if applicable, any specific attributes of a Subscriber. The results of this service are passed to the Certificate Generation Service.

- Certificate Application

  Certificate application is carried out by the Registration Service after identification of the Subscriber has been carried out meeting the requirements specified in the associated Certificate Policy.

- Subscriber Data Management

  The Registration Service by its nature must manage end entity subscriber data. The data may be affected by many different data protection requirements.

Certificate Generation Service: Creates and signs certificates based on the identity and other attributes of a Subscriber as verified by the Registration Service.

- Certificate Generation

After receiving a certificate application from the Registration Service, KTSs generate a certificate using the public key supplied. This ensures the CSP has locked the binding of the Subscriber's public key to its identity. KTSs may also send their Infrastructure[2] or Control[3] Public Keys to be certified by the Certificate Generation Service. This produces Infrastructure or Control Certificates.

Following Certificate Generation, the certificate is delivered to the Subscriber directly, and additionally it may be distributed via the Certificate Dissemination Service (publication of the certificate in a Directory).

Infrastructure and Control Certificates may be provided directly to the trustworthy component requiring its use.

- Certificate Renewal

  During the period prior to the expiration of the certificate, such period being defined by applicable policy, the certificate may be renewed. The certificate renewal consists of the following re-key (a new public key is certified using the registration information used to generate the previous certificate) and renewal scenario (the current public key is again certified). Certificate renewal covers Infrastructure, Control and Subscriber Certificates.

- Private Key Backup

  Subscriber private keys can be backed-up by the Certification Authority. Recovery of these keys shall be controlled by a multi-person principle as stated in this document.

Revocation Management Service: Processes requests and reports relating to revocation to determine the necessary action to be taken. The results of this service are distributed through the Revocation Status Service.

- Certificate Status Change Requests

  Where a Subscriber determines their private key may be compromised, a request for suspension (temporary revocation) of their certificate is sent to their CSP's KTS. A corresponding request to restore a certificate from suspension to operational use may be made by the Subscriber.

  Where the Subscriber knows for certain the private key is compromised, a request for revocation of their certificate is sent to their CSP's KTS.

  The CSP may also request a certificate status change via this service. Status of Control and Infrastructure Certificates may also be controlled through this service. Requests for certificate status change are authenticated messages and may be accepted or rejected by the CSP.

- Certificate Suspension/Revocation

[2] *Infrastructure keys are used by the some TOE components for processes such as subsystem authentication, audit log signing, encrypting transmitted, …*
[3] *Control keys are used by personnel managing or using the TOE components, and that may provide authentication, signing or confidentiality services for those personnel interacting with the system.*

The KTS having obtained a suspension or revocation request via this service changes the certificate status to either Suspended or Revoked (Figure 2-1: message A) in its Certificate Status Database, and this in turn is used by the CSP's Revocation Status Service.

Revocation Status Service: Provides certificate revocation status information to relying parties. This service is based on revocation status information that is updated at regular intervals.

- Revocation Status Data

  The Revocation Status Service provides certificate revocation status information to Relying Parties. The Revocation Status Service reflects changes to certificate status as status change requests either by the Subscriber or by the CSP are processed by the Revocation Management Service. This data may also be made available to Subscribers if policy requires Subscribers to have access to revocation status data.

- Status Request/Response

  A Relying Party having obtained the certificate(s) from the Certificate Dissemination Service, required for signature verification, needs to check the status of these certificates. The CSP provides a Revocation Status Service for this purpose. In the KeyOne system architecture the Revocation Status Service is an "online" service using Periodical messaging between the Revocation Status Service and the Revocation Management Service.

  In this "online" service, a Relying Party communicates with this Revocation Status Service and provides details of the certificate(s) for which status is required. The "online" Revocation Status Service when using Periodical messaging, queries its internal records, which have been updated by the last Periodical message. A reply is thus created and sent to the Relying Party indicating the status of the requested certificate(s).

  Figure 2-1 shows the relationship between the Revocation Management Service and the Revocation Status Service. In the figure, message A updates the TOE Certificate Status Database whereas Message B is a query/response message.

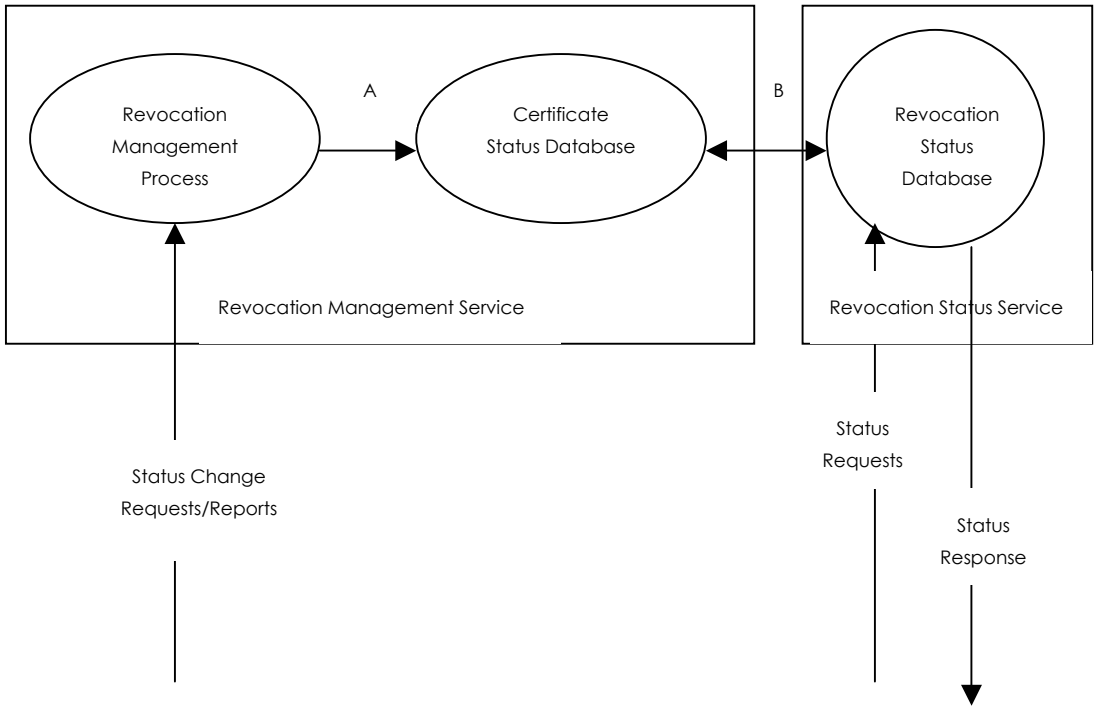

Figure 2-1. Messaging between Revocation Management Service and Revocation Status Service.

2.1.2 TOE Additional Services

Identified as optional in [CEN01c], the KTS also provides the following additional services:

Subscriber Device Provision Service: Prepares and provides a Signature Creation Device (SCD) to Subscribers

It is important to note that this service may provide a SCD and/or a SSCD. Within this ST the security requirements applicable to SCDs are equally applicable to SSCDs, where SSCDs meet the additional requirements stated in Annex III of [Eur99b].

- SCD Preparation

  The CSP's KTS prepares the SCD by performing the necessary initialisation, formatting and file structure creation.

  The KTS commands the SCD to generate the key pair inside the SCD.

- SCD Provision

  SCD Provision is the distribution of the SCD (after preparation) to the Subscriber.

- Activation Data Creation & Distribution

  The SCD is protected with (secret) activation data to protect the SCD contents. The CSP is responsible for generation of this initial activation data and subsequent secure distribution of this to the subscriber.

Time Stamp Service: A third party, trusted to provide a Time Stamp Service. The Time Stamp Service provides proof that a data item existed before a certain point in time (proof of existence). If the data item has been signed by the requester before being submitted to the Time Stamp Authority (TSA), then the Time Stamp Service provides proof that the data item existed and was signed before a certain point in time.

- Check Request Correctness

  This component is designed to check the correctness and the completeness of the request. If the result is positive, the data item is sent as input to the Time Stamp Generation.

- Time Parameter Generation

  This component uses a reliable source to deliver accurate time parameters. These parameters are used as input in the Time Stamp Generation process.

- Time Stamp Generation

  This function is responsible for creating a time stamp by binding the current time, a unique serial, the data provided for time stamping and ensuring any policy requirements are adhered to.

- Time Stamp Token (TST)

  This component is aimed at computing the time stamp token that is returned to the client. It effectively cryptographically signs the data provided by the Time Stamp Generation function.

The time stamping service,cryptographically binds time values to data values. The Figure 2-2 shows a conceptual TSA providing the time stamping service.

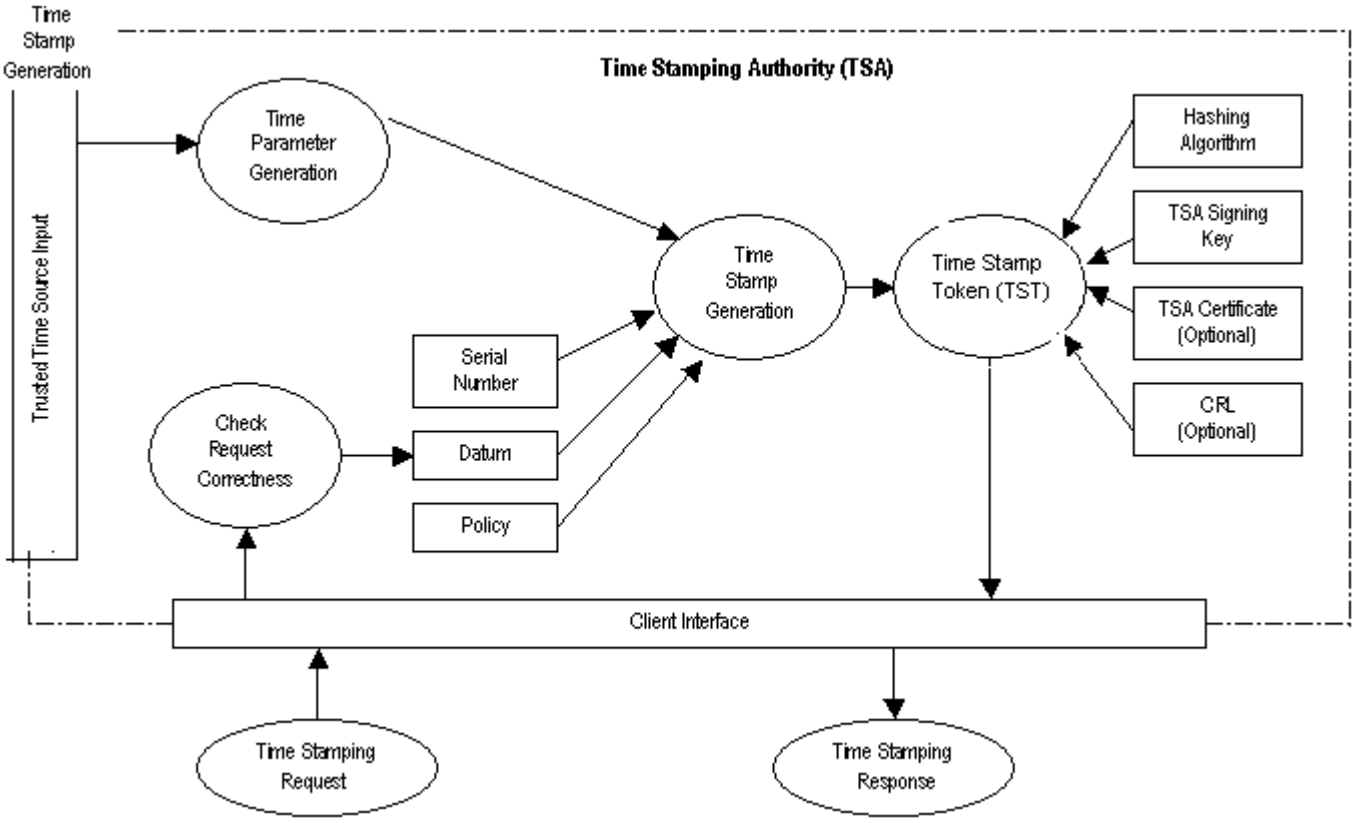

Figure 2-2. Time Stamping Services

2.1.3 TOE Users

The intended users of the TOE services are classified in two main groups:

2.1.3.1 External Users

End User / Certificate Entity, is the subject of the certificate that binds its identity with its public key. There are other types of entities that can be certified, for example applications, servers,

Relying Parties, users or agents or any external trust services that rely on the data in a certificate in making decisions, following the verification processes and limitations established in the certificate policies for each type of certificates issued by KTS.

Auditing Entities, who require access to audit trails for conducting evaluation processes to review certificate practices.

2.1.3.2 Internal Users

PKI administrators, who can configure and administer the different applications supported by the TOE: Registration, Certificate Authority, Validation Authority, Time Stamping authority.

Registration Officer is responsible for the operation of the Lightweight Registration Authority and the Registration Authority, according to established registration procedures

2.1.4 Overall Architecture

The Certification Service Provider (CSP) Services are shown in Figure 2-3 , and can be seen to facilitate the production and use of a signed transaction from the Subscriber to a Relying Party. This figure illustrates the services along with the TOE's interfaces to its Subscribers and Relying Parties.

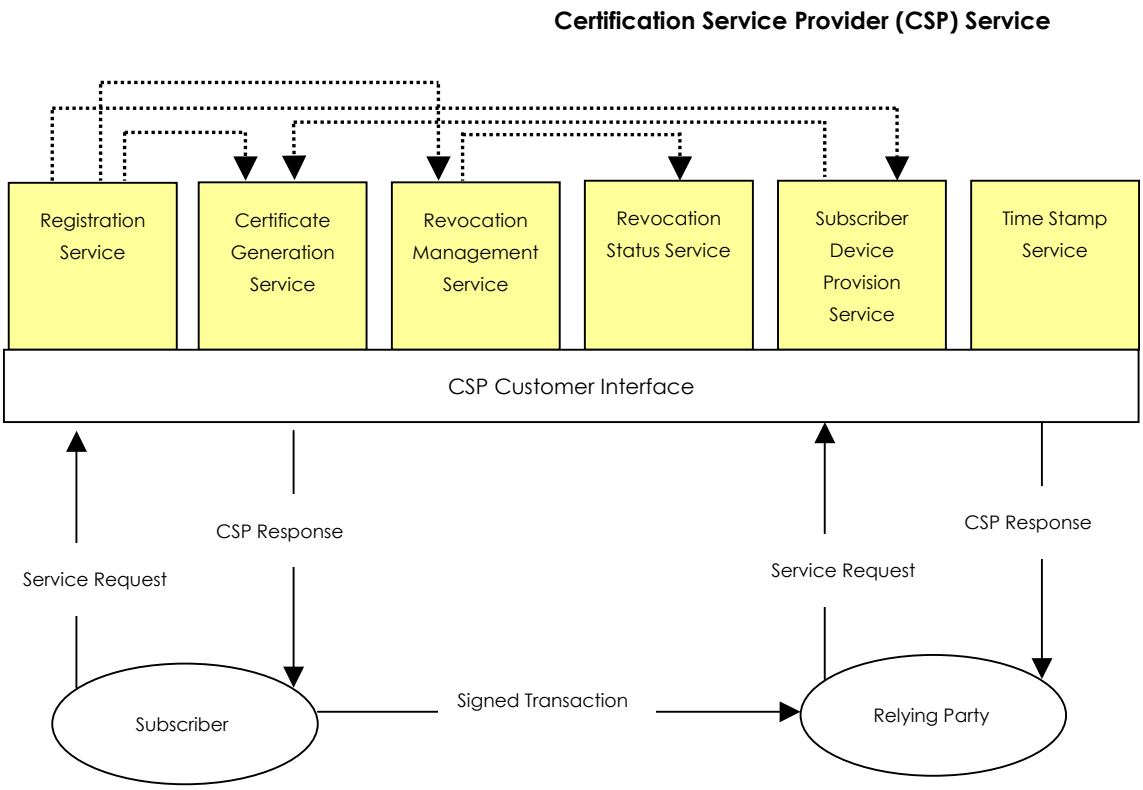

Figure 2-3. Certification Service Provider (CSP) Service

As shown, the TOE provides both initial registration and certificate generation. Primary certificate lifecycle management (where no revoked or suspended states exist) is provided by way of the Registration and Certificate Generation. Secondary certificate lifecycle management, where exceptional certificate states exist (e.g. revoked or suspended states) are provided by the Revocation Management and Revocation Status Services. The TOE Customer Interface provides access to the TOE's services by Subscribers and Relying Parties.

2.1.5 Logical Architecture

The logical architecture of the KTS is shown in the figure below.

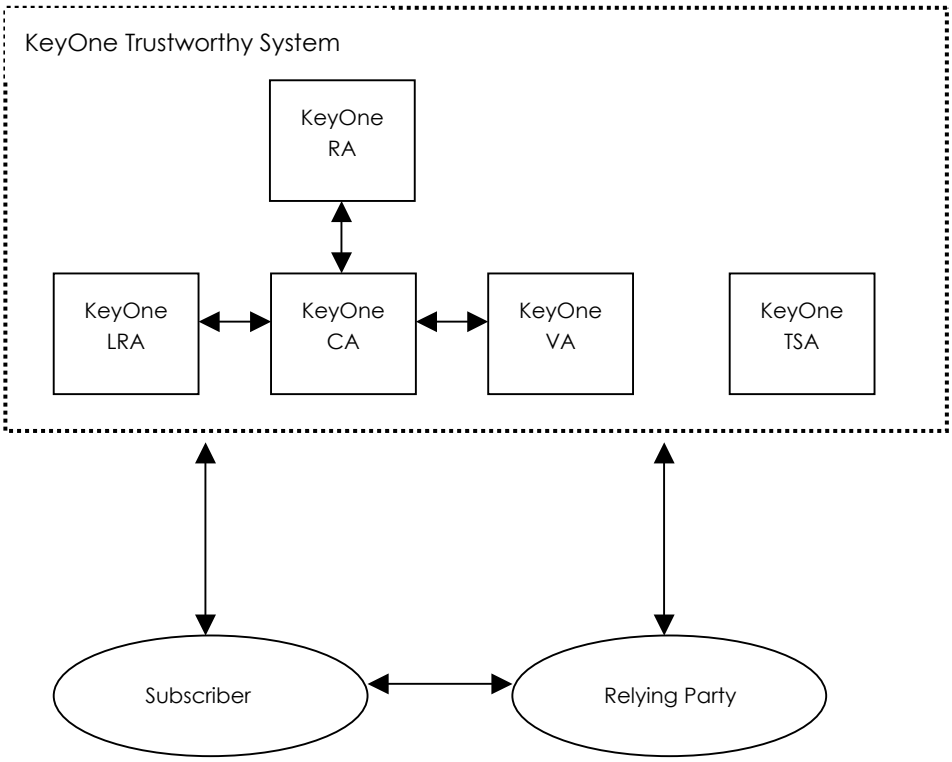

Figure 2-4. KTS logical architecture

Thus, the KeyOne system consists of the following elements:

- KeyOne LRA. The service related to this KeyOne component is explained in the **Registration Service** section, page 6, and in the **Subscriber Device Provision Service**, page 9
- KeyOne RA. The service related to this KeyOne component is explained in the **Registration Service** section, page 6.
- KeyOne CA. The service related to this KeyOne component is explained in the **Certificate Generation Service**, page 6 and in the **Revocation Management Service**, page 7.
- KeyOne VA. The service related to this KeyOne component is explained in the **Revocation Status Service**, page 8.
- KeyOne TSA. The service related to this KeyOne component is explained in the **Time Stamp Service** section, page 10.

2.1.6 Supported Services

This table enumerates the services supported by the system, and relates them with the KeyOne subsystem where them resides

Subsystem	Services

KeyOne LRA	Registration Service Subscriber Device Provision Service
KeyOne RA	Registration Service
KeyOne CA	Certificate Generation Service Revocation Management Service
KeyOne VA	Revocation Status Service
KeyOne TSA	Time Stamping Service

Table 2-1. Services supported by the system

Registration, Certificate Generation and Revocation Management Services and Subscriber Device Provision Service using the KeyOne LRA component

A subscriber of the certification service makes a certification request to KeyOne LRA, which, after verifying the subscriber's identity, redirects the request to KeyOne CA. It is actually KeyOne CA which then performs the requested certification and generates a response message that is sent back to KeyOne LRA. Once the intended certificate has been issued, it is included in the response message, so that KeyOne LRA can deliver it to the subscriber in a smartcard (SCD) using the Subscriber Device Provision Service. This service prepares (by generating the key pair inside the SC) and provides the SCD in order the could be delivered to the subscriber.

A subscriber of the revocation service makes a suspension, un-suspension or revocation request to KeyOne LRA, which, after verifying the subscriber's identity, redirects the request to KeyOne CA. It is actually KeyOne CA which performs then the requested revocation or suspension process and generates a response message that is sent back to KeyOne LRA.

Registration, Certificate Generation and Revocation Management Services using the KeyOne RA component

A subscriber of the certification service (operator) makes a certification request to KeyOne RA from the data of a subscriber of a certificate. When the certification request has been introduced, then other subscriber of the certification service (approver) can recover it and after verifying that the information contained in the request is correct, then he can approve the recovered request. When the request is approved, it is sent to the KeyOne CA through an internal KeyOne server. It is actually KeyOne CA which then performs the requested certification and generates a response message that is sent back to KeyOne RA. Once the intended certificate has been issued, it is included in the response message, so that KeyOne RA can deliver it to the subscriber.

A subscriber of the revocation service (operator) makes a suspension, un-suspension or revocation request to KeyOne RA. When the revocation request has been introduced, then other subscriber of the certification service (approver) can recover it and after verifying that the information contained in the request is correct, then he can approve the recovered request. When the request is approved, it is sent to KeyOne CA. It is actually KeyOne CA which performs then the requested revocation

or suspension process and generates a response message that is sent back to KeyOne RA.

KeyOne LRA/KeyOne RA – KeyOne CA Transactions

Registration, certificate generation and revocation services involve communication between KeyOne LRA/KeyOne RA and KeyOne CA. Messages exchanged during this communication process are called batches and fulfil an specific syntax, which includes a digital signature.

Furthermore, these messages are transferred over an SSL connection. Therefore, confidentiality, authenticity and integrity of KeyOne LRA/KeyOne RA - KeyOne CA transactions are guaranteed.

Batches can be classified in two categories, depending on the type of request they come from:

- CR batches: Batches that contain a certification request.
- RR batches: Batches that contain a revocation, suspension or un-suspension request.

Revocation Status Service

A subscriber of the revocation status service sends an OCSP request to KeyOne VA Server to determine the revocation state of a certain certificate. KeyOne VA, in turn, generates an OCSP response message after consulting its internal database and sends it back to the subscriber, who must proceed accordingly on receiving the response. Moreover, both the incoming and the outgoing OCSP messages are logged into the KeyOne VA internal database, so that later audits are possible.

Time Stamping Service

A subscriber of the time stamping service computes the digital fingerprint of some data and submits a time stamp request to KeyOne TSA, according to the TSP syntax defined in RFC 3161. Then, KeyOne TSA obtains the current time from a secure clock and binds both the data fingerprint and the moment when it was performed by issuing a signed time stamp token. Finally, the timestamp token is encapsulated in a TSP response message and sent back to the subscriber.

The issued time stamp token is a proof of existence of the time stamped data, that is, an unforgeable evidence that the data existed before a certain point in time. Both the incoming and the outgoing OCSP messages are logged into the KeyOne TSA internal database, so that later audits and verifications are allowed.

2.1.7 Physical Architecture

The physical architecture of the KTS is shown in the figure below:

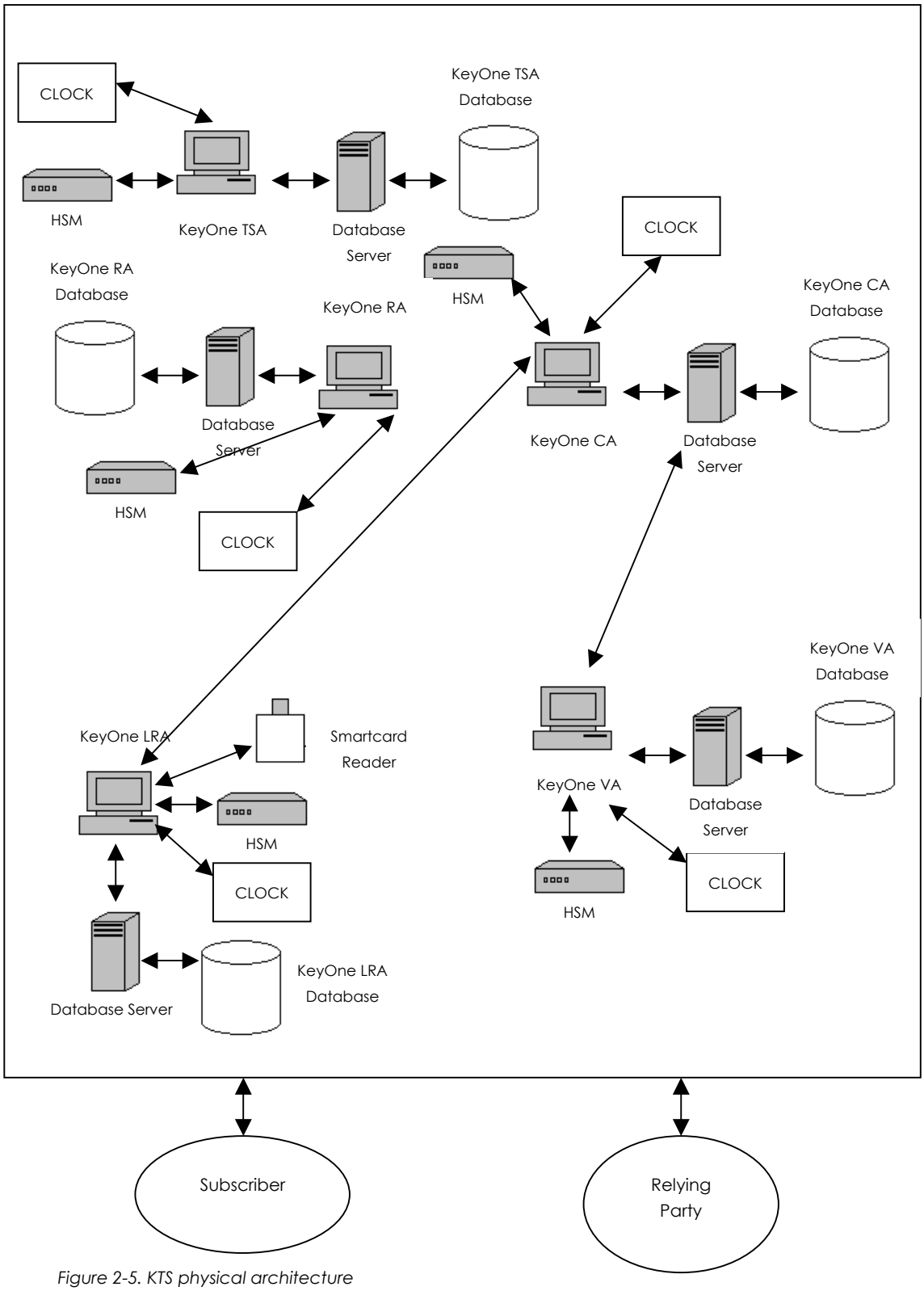

Figure 2-5. KTS physical architecture

This figure shows the components included in the physical architecture of the KTS.

All KeyOne components are connected to a Database where the information related to the service that the component provides is stored. The database related to the KeyOne CA component stores generated certificates and CRLs, KeyOne batches (batches contain sets of certification or revocation requests, or certificates, depending on the entity that issued it. The main purpose of batches used in KeyOne is to send certification or revocation requests and to receive responses between the RA and the CA) and logs generated by the KeyOne CA subsystem. The database related to the KeyOne VA component stores the status related to the certificates, the messages interchanged with the KeyOne CertStatus component (part of the KeyOne CA product), and the logs generated by the KeyOne VA subsystem. The database related to the KeyOne TSA component stores TSTs requests and responses, and the logs generated by the KeyOne TSA subsystem. The database related to the KeyOne RA component stores certificates, KeyOne batches and logs generated by the KeyOne RA subsystem. The database related to the KeyOne LRA component stores logs generated by the KeyOne LRA subsystem.

All KeyOne components are connected to an HSM (Hardware Security Module) in order to generate and store the keys related to the service, and also they are connected to a clock that provides reliable time stamps for the service use.

In the KeyOne LRA component, a user can request the generation of a certificate (generation of the keys in a smartcard) or the status change of a previously generated certificate. In this case, a Registration Operator verifies the identity of the requesting entity, and approves or denies the request signing it with his signature certificate stored in a smartcard. When the Registration Operator signs the request, then it is sent inside a KeyOne batch to the KeyOne CA Server; this server processes the request (changes the status of the certificate in the KeyOne CA database or generates the certificate) and sends the result to the KeyOne LRA Server inside a KeyOne batch. If the request implies the generation of a certificate, then it will be stored in the subscriber smartcard.

In the KeyOne RA component, an operator can request the generation of a certificate or the status change of a previously generated certificate. In this case, the request is introduced, and after an approver operator verifies the introduced request and approves or denies the request signing it with his signature. When the Registration Operator signs the request, then it is sent inside a KeyOne batch to the KeyOne CA Server; this server processes the request (changes the status of the certificate in the KeyOne CA database or generates the certificate) and sends the result to the KeyOne RA Server inside a KeyOne batch.

KeyOne CA (KeyOne CertStatus Server) accesses to KeyOne CA database, where the information on certificate revocation status is stored. Every now and then, KeyOne VA will send requests to KeyOne CA (KeyOne CertStatus Server) to obtain the list of certificates that have changed their status in the last time lapse. NDCCP (Near Domain Cert-status Coverage Protocol) is a KeyOne proprietary protocol, which is used in the communication between a Database Updater module (in KeyOne VA) and a CertStatus Server module (in KeyOne CA).

KeyOne VA (Validation Authority) Server implements the Online Certificate Status Protocol (OCSP) and determines the current status of a digital certificate. Using this service, a relying party requests to the Validation Authority for the status of a certificate.

The environment components for each of the KTS components are listed in the table below.

Subsystem	OS	Database	HSM	SCD/SSCD
KeyOne CA	Microsoft Windows 2000	Oracle 9i	nCipher nShield F3 Ultrasign SCSI, firmware version 2.0.5, Hardware version nC4032W	STARCOS SPK 2.4
KeyOne LRA	Microsoft Windows 2000	Oracle 9i	nCipher nShield F3 Ultrasign SCSI, firmware version 2.0.5, Hardware version nC4032W	STARCOS SPK 2.4
KeyOne RA	Microsoft Windows 2000	Oracle 9i	nCipher nShield F3 Ultrasign SCSI, firmware version 2.0.5, Hardware version nC4032W	STARCOS SPK 2.4
KeyOne VA	Microsoft Windows 2000	Oracle 9i	nCipher nShield F3 Ultrasign SCSI, firmware version 2.0.5, Hardware version nC4032W	STARCOS SPK 2.4
KeyOne TSA	Microsoft Windows 2000	Oracle 9i	nCipher nShield F3 Ultrasign SCSI, firmware version 2.0.5, Hardware version nC4032W	STARCOS SPK 2.4

Table 2-2. Environment components

Additionally, it is necessary the following components:

- NTP client installed in the same host where the KeyOne CA, KeyOne RA, KeyOne LRA, KeyOne TSA and KeyOne VA subsystems.
- Reliable clock that obtains the Co-ordinated Universal Time from a reliable source, and that synchronizes the system clock by means the NTP protocol, using the NTP client installed in the same machine that the KeyOne CA, KeyOne RA, KeyOne LRA, KeyOne TSA and KeyOne VA subsystems.
- Windows 2000 Service Pack 4 for the KeyOne CA, KeyOne LRA, KeyOne RA, KeyOne VA and KeyOne TSA components.

2.2 Use Cases

The TOE functionality presented in this document may solve a great variety of business cases, reaching from user identification and controlling the access to internal resources, to electronic commerce, and many diverse sectors of the market.

However, the security services that KTS provides can be summarized in the following:

1 User / Entity / Application authentication

2 Information encryption

3 Non-repudiation and integrity, provided by advanced digital signature.

Therefore, whatever business or application that requires any of the before mentioned security services, is capable of using a KTS.

The use of this TOE is more suitable to certain registry schemes. This configuration adapts perfectly to:

- Distributed register environments, due to the fast and easy deployment of different RAs, without increasing maintenance needs.
- Mobile or travelling registers, guaranteeing the security of the register service without very restrictive physical protections measures.

CHAPTER 3

3 TOE Security Environment

This section includes the following:

- Secure usage assumptions,
- Threats, and
- Organizational security policies.

This information provides the basis for the Security Objectives specified in Section 4, the security functional requirements for the TOE and environment specified in Sections 5 and 6, and the TOE Security Assurance Requirements specified in Section 8.

3.1 Secure Usage Assumptions

The usage assumptions are organized in three categories: personnel (assumptions about administrators and users of the system as well as any threat agents), physical (assumptions about the physical location of the TOE or any attached peripheral devices), and connectivity (assumptions about other IT systems that are necessary for the secure operation of the TOE).

3.1.1 Personnel

A.Auditors Review Audit Logs

Audit logs are required for security-relevant events and must be reviewed by the Auditors.

A.Authentication Data Management

An authentication data management policy is enforced to ensure that users change their authentication data at appropriate intervals and to appropriate values (e.g., proper lengths, histories, variations, etc.) (Note: this assumption is not applicable to biometric authentication data.)

A.Competent Administrators, Operators, Officers and Auditors

Competent Administrators, Operators, Officers and Auditors will be assigned to manage the TOE and the security of the information it contains.

A.CPS

All Administrators, Operators, Officers, and Auditors are familiar with the certificate policy (CP) and certification practices statement (CPS) under which the TOE is operated. This documentation is conformant with [NPKI] certificate policy.

A.Disposal of Authentication Data

Proper disposal of authentication data and associated privileges is performed after access has been removed (e.g., job termination, change in responsibility).

A.Malicious Code Not Signed

Malicious code destined for the TOE is not signed by a trusted entity.

A.Notify Authorities of Security Issues

Administrators, Operators, Officers, Auditors, and other users notify proper authorities of any security issues that impact their systems to minimize the potential for the loss or compromise of data.

A.Social Engineering Training

General users, administrators, operators, officers and auditors are trained in techniques to thwart social engineering attacks.

A.Cooperative Users

Users need to accomplish some task or group of tasks that require a secure IT environment. The users require access to at least some of the information managed by the TOE and are expected to act in a cooperative manner.

3.1.2 Connectivity

A.Operating System

The operating system has been selected to provide the functions required by this CIMC to counter the perceived threats for the CIMC level 3 PP, as identified in this Security Target.

A.NTP Client

All the hosts included in the TOE have installed an NTP client that synchronises the system clock with a reliable clock that obtains the Coordinated Universal Time from a reliable source.

3.1.3 Physical

A.Communications Protection

The system is adequately physically protected against loss of communications i.e., availability of communications.

A.Physical Protection

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

3.2 Threats

The threats are organized in four categories: authorized users, system, cryptography, and external attacks.

3.2.1 Authorized Users

T.Administrative errors of omission

Administrators, Operators, Officers or Auditors fail to perform some function essential to security.

T.User abuses authorization to collect and/or send data

User abuses granted authorizations to improperly collect and/or send sensitive or security-critical data.

T.User error makes data inaccessible

User accidentally deletes user data rendering user data inaccessible.

T.Administrators, Operators, Officers and Auditors commit errors or hostile actions

An Administrator, Operator, Officer or Auditor commits errors that change the intended security policy of the system or application or maliciously modify the system's configuration to allow security violations to occur.

3.2.2 System

T.Critical system component fails

Failure of one or more system components results in the loss of system critical functionality.

T.Malicious code exploitation

An authorized user, IT system, or hacker downloads and executes malicious code, which causes abnormal processes that violate the integrity, availability, or confidentiality of the system assets.

T.Message content modification

A hacker modifies information that is intercepted from a communications link between two unsuspecting entities before passing it on to the intended recipient.

T.Flawed code

A system or applications developer delivers code that does not perform according to specifications or contains security flaws.

3.2.3 Cryptography

T.Disclosure of private and secret keys

A private or secret key is improperly disclosed.

T.Modification of private/secret keys

A secret/private key is modified.

T.Sender denies sending information

The sender of a message denies sending the message to avoid accountability for sending the message and for subsequent action or inaction.

3.2.4 External Attacks

T.Hacker gains access

A hacker masquerades as an authorized user to perform operations that will be attributed to the authorized user or a system process or gains undetected access to a system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality, or availability.

T.Hacker physical access

A hacker physically interacts with the system to exploit vulnerabilities in the physical environment, resulting in arbitrary security compromises.

T.Social engineering

A hacker uses social engineering techniques to gain information about system entry, system use, system design, or system operation.

3.3 Organizational Security Policies

P.Authorized use of information

Information shall be used only for its authorized purpose(s).

P.Cryptography

FIPS-approved or NIST-recommended cryptographic functions shall be used to perform all cryptographic operations.

CHAPTER 4

4 Security Objectives

This section identifies and defines the security objectives for the TOE and its environment. Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organisational security policies and assumptions.

4.1 Security Objectives for the TOE

This section includes the security objectives for the TOE, divided among four categories: authorized users, system, cryptography, and external attacks.

4.1.1 Authorized Users

O.Certificates

The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.

4.1.2 System

O.Preservation/trusted recovery of secure state

Preserve the secure state of the system in the event of a secure component failure and/or recover to a secure state.

O.Sufficient backup storage and effective restoration

Provide sufficient backup storage and effective restoration to ensure that the system can be recreated.

4.1.3 Cryptography

O.Non-repudiation

Prevent user from avoiding accountability for sending a message by providing evidence that the user sent the message.

4.1.4 External Attacks

O.Control unknown source communication traffic

Control (e.g., reroute or discard) communication traffic from an unknown source to prevent potential damage.

4.2 Security Objectives for the Environment

This section specifies the security objectives for the environment.

4.2.1 Non-IT security objectives for the environment

O.Administrators, Operators, Officers and Auditors guidance documentation

Deter Administrator, Operator, Officer or Auditor errors by providing adequate documentation on securely configuring and operating the CIMC.

O.Auditors Review Audit Logs

Identify and monitor security-relevant events by requiring auditors to review audit logs on a frequency sufficient to address level of risk.

O.Authentication Data Management

Ensure that users change their authentication data at appropriate intervals and to appropriate values (e.g., proper lengths, histories, variations, etc.) through enforced authentication data management (Note: this objective is not applicable to biometric authentication data.)

O.Communications Protection

Protect the system against a physical attack on the communications capability by providing adequate physical security.

O.Competent Administrators, Operators, Officers and Auditors

Provide capable management of the TOE by assigning competent Administrators, Operators, Officers and Auditors to manage the TOE and the security of the information it contains.

O.CPS

All Administrators, Operators, Officers and Auditors shall be familiar with the certificate policy (CP) and the certification practices statement (CPS) under which the TOE is operated.

O.Disposal of Authentication Data

Provide proper disposal of authentication data and associated privileges after access has been removed (e.g., job termination, change in responsibility).

O.Installation

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security.

O.Malicious Code Not Signed

Protect the TOE from malicious code by ensuring all code is signed by a trusted entity prior to loading it into the system.

O.Notify Authorities of Security Issues

Notify proper authorities of any security issues that impact their systems to minimize the potential for the loss or compromise of data.

O.Physical Protection

Those responsible for the TOE must ensure that the security-relevant components of the TOE are protected from physical attack that might compromise IT security.

O.Social Engineering Training

Provide training for general users, Administrators, Operators, Officers and Auditors in techniques to thwart social engineering attacks.

O.Cooperative Users

Ensure that users are cooperative so that they can accomplish some task or group of tasks that require a secure IT environment and information managed by the TOE. .

O.Lifecycle security

Provide tools and techniques used during the development phase to ensure security is designed into the CIMC. Detect and resolve flaws during the operational phase.

O.Repair identified security flaws

The vendor repairs security flaws that have been identified by a user.

4.2.2 IT security objectives for the environment

O.Cryptographic functions

The TOE must implement approved cryptographic algorithms for encryption/decryption, authentication, and signature generation/verification; approved key generation techniques and use validated cryptographic modules. (Validated is defined as FIPS 140-2 validated.)

O.Operating System

The operating system used is validated to provide adequate security, including domain separation and non-bypassability, in accordance with security requirements recommended by the National Institute of Standards and Technology.

O.Periodically check integrity

Provide periodic integrity checks on both system and software.

O.Security roles

Maintain security-relevant roles and the association of users with those roles.

O.Validation of security function

Ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures.

O.Trusted Path

Provide a trusted path between the user and the system. Provide a trusted path to security-relevant (TSF) data in which both end points have assured identities.

4.3 Security Objectives for both the TOE and the Environment

This section specifies the security objectives that are jointly addressed by the TOE and the environment.

O.Configuration Management

Implement configuration management to assure identification of system connectivity (software, hardware, and firmware), and components (software, hardware, and

firmware), auditing of configuration data, and controlling changes to configuration items.

O.Data import/export

Protect data assets when they are being transmitted to and from the TOE, either through intervening untrusted components or directly to/from human users.

O.Detect modifications of firmware, software, and backup data

Provide integrity protection to detect modifications to firmware, software, and backup data.

O.Individual accountability and audit records

Provide individual accountability for audited events. Record in audit records: date and time of action and the entity responsible for the action.

O.Integrity protection of user data and software

Provide appropriate integrity protection for user data and software.

O.Limitation of administrative access

Design administrative functions so that Administrators, Operators, Officers and Auditors do not automatically have access to user objects, except for necessary exceptions. Control access to the system by Operators and Administrators who troubleshoot the system and perform system updates.

O.Maintain user attributes

Maintain a set of security attributes (which may include role membership. access privileges, etc.) associated with individual users. This is in addition to user identity.

O.Manage behavior of security functions

Provide management functions to configure, operate, and maintain the security mechanisms.

O.Object and data recovery free from malicious code

Recover to a viable state after malicious code is introduced and damage occurs. That state must be free from the original malicious code.

O.Procedures for preventing malicious code

Incorporate malicious code prevention procedures and mechanisms.

O.Protect stored audit records

Protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.

O.Protect user and TSF data during internal transfer

Ensure the integrity of user and TSF data transferred internally within the system.

O.Require inspection for downloads

Require inspection of downloads/transfers.

O.Respond to possible loss of stored audit records

Respond to possible loss of audit records when audit trail storage is full or nearly full by restricting auditable events.

O.Restrict actions before authentication

Restrict the actions a user may perform before the TOE authenticates the identity of the user.

O.Security-relevant configuration management

Manage and update system security policy data and enforcement functions, and other security-relevant configuration data, to ensure they are consistent with organizational security policies.

O.Time stamps

Provide time stamps to ensure that the sequencing of events can be verified.

O.User authorization management

Manage and update user authorization and privilege data to ensure they are consistent with organizational security and personnel policies.

O.React to detected attacks

Implement automated notification (or other responses) to the TSF-discovered attacks in an effort to identify attacks and to create an attack deterrent.

5 IT Security Requirements

Some requirements in this chapter reference to the following roles: Administrator, Operator, Officer and Auditor. These roles have been extracted from the CIMC Protection Profile, and the definitions for these roles are listed in the section "5.2. Roles" of the [CIMC] document.

5.1 TOE Security Requirements

5.1.1 TOE Security Functional Requirements

The required minimum strength of function level is mandated as "SOF-basic", for the functional requirements indicated in this section. With this SOF, the TOE shall be resistant to attackers with low attack potential, and remaining vulnerabilities shall only be exploitable by attacker with moderate or high attack potential.

This section specifies the security functional requirements that are applicable to the TOE. All these requirements has been extracted from the [CIMC] Protection Profile. Some of these requirements has been instantiated by means the use of the operations mechanism offered by the Common Criteria. The following table lists all the security functional requirements for the TOE, and the type of operation applied to them.

Functional Requirement	*Security Target Operation*
FAU_GEN.1.1 (FAU_GEN.1 iteration 2)	Selection, Assignment[4]
FAU_GEN.1.2 (FAU_GEN.1 iteration 2)	Refinement, Assignment
FAU_GEN.2.1 (FAU_GEN.2 iteration 2)	None
FAU_SEL.1.1 (FAU_SEL.1 iteration 2)	Selection, Assignment
FAU_STG.1.1 (FAU_STG.1 iteration 2)	None
FAU_STG.1.2 (FAU_STG.1 iteration 2)	Selection
FAU_STG.4.1 (FAU_STG.4 iteration 2)	Assignment, Selection
FPT_STM.1.1 (FPT_STM.1 iteration 2)	None

[4] *Regarding to the CIMC Protection Profile, a refinement operation has been applied.*

FMT_MOF.1.1 (FMT_MOF.1 iteration 2)	Assignment, Selection
FDP_ACC.1.1 (FDP_ACC.1 iteration 2)	Assignment
FDP_ACF.1.1 (FDP_ACF.1 iteration 2)	Assignment
FDP_ACF.1.2 (FDP_ACF.1 iteration 2)	Assignment
FDP_ACF.1.3 (FDP_ACF.1 iteration 2)	Assignment
FDP_ACF.1.4 (FDP_ACF.1 iteration 2)	Assignment
FDP_ITT.1.1 (FDP_ITT.1 iteration 3)	Assignment, Selection
FDP_ITT.1.1 (FDP_ITT.1 iteration 4)	Assignment, Selection
FDP_UCT.1.1 (FDP_UCT.1 iteration 2)	Assignment, Selection
FPT_RVM.1.1 (FPT_RVM.1 iteration 2)	None
FPT_ITC.1.1 (FPT_ITC.1 iteration 2)	Refinement
FPT_ITT.1.1 (FPT_ITT.1 iteration 3)	Selection, Refinement
FPT_ITT.1.1 (FPT_ITT.1 iteration 4)	Selection, Refinement
FIA_UAU.1.1 (FIA_UAU.1 iteration 2)	Assignment
FIA_UAU.1.2 (FIA_UAU.1 iteration 2)	None
FIA_UID.1.1 (FIA_UID.1 iteration 2)	Assignment
FIA_UID.1.2 (FIA_UID.1 iteration 2)	None
FIA_USB.1.1 (FIA_USB.1 iteration 2)	None
FPT_CIMC_TSP.1.1	None
FPT_CIMC_TSP.1.2	None
FPT_CIMC_TSP.1.3	None
FPT_CIMC_TSP.1.4	None
FDP_ACF_CIMC.2.1	None
FDP_ACF_CIMC.2.2	None
FDP_ACF_CIMC.3.1	None
FDP_SDI_CIMC.3.1	None
FDP_SDI_CIMC.3.2	Assignment
FDP_ETC_CIMC.5.1	None
FDP_CIMC_BKP.1.1	None
FDP_CIMC_BKP.1.2	None
FDP_CIMC_BKP.1.3	None
FDP_CIMC_BKP.1.4	None
FDP_CIMC_BKP.2.1	None

FDP_CIMC_BKP.2.2	None
FDP_CIMC_CSE.1.1	Assignment
FDP_CIMC_CER.1.1	Assignment
FDP_CIMC_CER.1.2	None
FDP_CIMC_CER.1.3	None
FDP_CIMC_CER.1.4	None
FDP_CIMC_CRL.1.1	None
FDP_CIMC_OCSP.1.1	None
FCO_NRO_CIMC.3.1	None
FCO_NRO_CIMC.3.2	Assignment
FCO_NRO_CIMC.3.3	None
FCO_NRO_CIMC.4.1	None
FCO_NRO_CIMC.4.2	None
FMT_MTD_CIMC.4.1	None
FMT_MTD_CIMC.5.1	None
FMT_MTD_CIMC.7.1	None
FMT_MOF_CIMC.3.1	None
FMT_MOF_CIMC.3.2	None
FMT_MOF_CIMC.3.3	None
FMT_MOF_CIMC.3.4	None
FMT_MOF_CIMC.5.1	None
FMT_MOF_CIMC.5.2	None
FMT_MOF_CIMC.5.3	None
FMT_MOF_CIMC.6.1	None
FMT_MOF_CIMC.6.2	None
FMT_MOF_CIMC.6.3	None
FCS_CKM_CIMC.5.1	None

Table 5-3. Functional Requirements for the TOE

5.1.1.1 FAU – Security audit

Security auditing involves recognizing, recording, storing and analyzing information related to security relevant activities (i.e. activities controlled by the TSP). The resulting

audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

Audit includes a chronological recording of events that occur in a system. The objective is to track what occurs to enable the reconstruction and examination of a sequence of events and/or changes in an event. This is useful in ensuring that the system is operated securely and in providing evidence when a suspected or actual security compromise has occurred. Audit also provides for reconstructing a specific state of a system. The objective in a PKI system is to enable an appropriate authority to determine whether a signature should have been accepted as valid.

The audit will be used to reconstruct important events that were performed by the TOE, such as issuance of a CA certificate, and the user or event (e.g., a signed certificate request) that caused them. The audit will be used to arbitrate future disputes by establishing the validity of a signature at a particular time.

The audit log records the security-relevant events that were performed by the TOE and the users or events (e.g., a signed certificate request) that caused them. This subsection specifies the security requirements for maintaining and protecting the integrity of the audit logs.

FAU_GEN – Security Audit Data Generation

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

FAU_GEN.1 Audit Data Generation (iteration 2)

Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

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 [*minimum*] level of audit; and

c) [

- *Any changes to the audit parameters, e.g., audit frequency, type of event audited. Any attempt to delete the audit log.*
- *Audit log signing event.*
- *All security-relevant data that is entered in the system in a Local Data Entry context. The Local Data Entry context implies that a user, operating locally, enters or accept data so that the system can associate the data with the user and list the user in the audit log with the accepted data*

(this data entry could take the form of a user vouching for information that has already been entered into the computer by clicking on an "accept" button or by otherwise indicating acceptance of the information).

- *All security-relevant messages that are received by the system in a Remote Data Entry context. The Remote Data Entry context implies that related data could be received over a network in such a way that it can be bound to the identity of the sender of the data (or to the identity of some other user).*
- *All successful and unsuccessful requests for confidential and security-relevant information in a Data Export and Output context.*
- *Whenever the TSF requests generation of a cryptographic key (not mandatory for single session or one-time use symmetric keys).*
- *The loading of Component private keys.*
- *All access to certificate subject private keys retained within the TOE for key recovery purposes.*
- *All changes to the trusted public keys, including additions and deletions.*
- *The manual entry of secret keys used for authentication.*
- *The export of private and secret keys (keys used for a single session or message are excluded).*
- *All certificate requests.*
- *All requests to change the status of a certificate.*
- *Any security-relevant changes to the configuration of the TSF.*
- *All changes to the certificate profile.*
- *All changes to the revocation profile.*
- *All changes to the certificate revocation list profile.*
- *All changes to the OCSP profile.]*

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, [*the following information:*

- *Digital signature, keyed hash, or authentication code shall be included in the audit log. This information will be recorded in the register of the audit log signing event.*
- *The identity of the data entry individual if the entered data is linked to any other data (e.g., clicking an "accept" button). This shall be*

included with the accepted data. This information will be recorded in the register of all security-relevant data that is entered in the system.

- *The public key component of any asymmetric key pair generated. This information will be recorded in the register of the TSF requests generation of a cryptographic key*
- *The public key and all information associated with the key (in operations of changes, additions and deletions of trusted public keys).*
- *The copy of the related certificate when a certificate request is accepted, and the reason for rejection when a certificate request is rejected.*
- *Whether a request to change the status of a certificate was accepted or rejected.*
- *The changes made to the profile, when a change in the certificate profile is requested.*
- *The changes made to the profile, when a change in the revocation profile is requested.*
- *The changes made to the profile, when a change in the certificate revocation list profile is requested.*
- *The changes made to the profile, when a change in the OCSP profile is requested.]*

Additionally, the audit shall not include plaintext private or secret keys or other critical security parameters.

FAU_GEN.2 User Identity Association (iteration 2)

The TSF shall associate auditable events to individual user identities.

FAU_GEN.2.1

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

FAU_SEL – Security Audit Event Selection

This family defines requirements to select the events to be audited during TOE operation. It defines requirements to include or exclude events from the set of auditable events.

FAU_SEL.1 Selective Audit (iteration 2)

Selective Audit, requires the ability to include or exclude events from the set of audited events based upon attributes to be specified by the PP/ST author.

FAU_SEL.1.1

The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) [selection: *event type*]

b) [assignment: *no additional attributes*]

FAU_STG – Security Audit Event Storage

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail.

FAU_STG.1 Protected audit trail storage (iteration 2)

Protected audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorised deletion and/or modification.

FAU_STG.1.1

The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2

The TSF shall be able to [*detect*] unauthorized modifications to the audit records in the audit trail.

FAU_STG.4 Prevention of audit data loss (iteration 2)

FAU_STG.4 Prevention of audit data loss specifies actions in case the audit trail is full.

FAU_STG.4.1

The TSF shall [*prevent auditable events, except those taken by Auditor*] and [assignment: *shuts down the system*] if the audit trail is full.

5.1.1.2 FPT – Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data).

FPT_STM – Time stamps

This family addresses requirements for a reliable time stamp function within a TOE.

FPT_STM.1 Reliable time stamps (iteration 2)

This component requires that the TSF provide reliable time stamps for TSF functions.

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps for its own use.

5.1.1.3 FMT – Security Management

This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

FMT_MOF – Management of functions in TSF

This family allows authorized users control over the management of functions in the TSF. Examples of functions in the TSF include the audit functions and the multiple authentication functions.

FMT_MOF.1 Management of security functions behavior (iteration 2)

This component allows the authorized users (roles) to manage the behavior of functions in the TSF that use rules or have specified conditions that may be manageable.

FMT_MOF.1.1

The TSF shall restrict the ability to [*modify the behaviour of*] the functions [*list of functions listed in the table below*] to [*the authorised roles as specified in the table below*]

Section/Function	Component	Function/Authorized Role
Security Audit		The capability to configure the audit parameters shall be restricted to Administrators.
Backup and Recovery		The capability to configure the backup parameters shall be restricted to Administrators. The capability to initiate the backup or recovery function shall be restricted to [assignment: *Administrator*]
Certificate Registration		The capability to approve fields or extensions to be included in a certificate shall be restricted to Officers. If an automated process is used to approve fields or extensions to be included in a certificate, the capability to configure that process shall be restricted to Officers.
Data Export and Output		The export of KTS private keys shall require the authorization of at least two Administrators or one Administrator and one Officer, Auditor, or Operator.

Certificate Status Change Approval		Only Officers shall configure the automated process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall configure the automated process used to approve the placing of a certificate on hold or information about the hold status of a certificate.
KTS Configuration		The capability to configure any TSF functionality shall be restricted to Administrators. (This requirement applies to all configuration parameters unless the ability to configure that aspect of the TSF functionality has been assigned to a different role elsewhere in this document).
Certificate Profile Management	FMT_MOF_CIMC.2 Certificate profile management FMT_MOF_CIMC.3 Extended certificate profile management	The capability to modify the certificate profile shall be restricted to Administrators.
Revocation Profile Management		The capability to modify the revocation profile shall be restricted to Administrators.
Certificate Revocation List Profile Management	FMT_MOF_CIMC.4 Certificate revocation list profile management FMT_MOF_CIMC.5 Extended certificate revocation list profile management	The capability to modify the certificate revocation list profile shall be restricted to Administrators.
Online Certificate Status Protocol (OCSP) Profile Management	FMT_MOF_CIMC.6 OCSP profile management	The capability to modify the OCSP profile shall be restricted to Administrators.

Table 5-1. Authorized Roles for Management of Security Functions Behavior

5.1.1.4 FDP – User Data Protection

This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. FDP is split into four groups of families (listed below) that address user data within a TOE, during import, export, and storage as well as security attributes directly related to user data.

FDP_ACC – Access control policy

This family identifies the access control SFPs (by name) and defines the scope of control of the policies that form the identified access control portion of the TSP. This scope of control is characterized by three sets: the subjects under control of the policy, the objects under control of the policy, and the operations among controlled subjects and controlled objects that are covered by the policy. The criteria allows multiple policies to exist, each having a unique name. This is accomplished by iterating components from this family once for each named access control policy.

The rules that define the functionality of an access control SFP will be defined by other families such as FDP_ACF and FDP_SDI. The names of the access control SFPs identified here in FDP_ACC are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an "access control SFP."

FDP_ACC.1 Subset access control (iteration 2)

This component requires that each identified access control SFP be in place for a subset of the possible operations on a subset of the objects in the TOE.

FDP_ACC.1.1

The TSF shall enforce the [*CIMC TOE Access Control Policy specified in chapter 2 of the SPM*] on [assignment: *All users of the application successfully identified and authenticated, configuration data, operations and function code , and access and code execution that can be assigned to the application roles*].

FDP_ACF – Access control functions

This family describes the rules for the specific functions that can implement an access control policy named in FDP_ACC. FDP_ACC specifies the scope of control of the policy.

This family addresses security attribute usage and characteristics of policies. The component within this family is meant to be used to describe the rules for the function that implements the SFP as identified in FDP_ACC. The PP/ST author may also iterate this component to address multiple policies in the TOE.

FDP_ACF.1 Security attribute based access control (iteration 2)

This component allows the TSF to enforce access based upon security attributes and named groups of attributes. Furthermore, the TSF may have the ability to explicitly authorize or deny access to an object based upon security attributes.

FDP_ACF.1.1

The TSF shall enforce the [*CIMC TOE Access Control Policy specified in chapter 2 of the SPM*] to objects based on the following: [*the identity of the subject and the set of roles that the subject is authorized to assume*].

FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*rules specified in the table below*].

Section/Function	Component	Function/Authorized Role
Certificate Request Remote and Local Data Entry		The entry of certificate request data shall be restricted to Officers and the subject of the requested certificate.
Certificate Revocation Request Remote and Local Data Entry		The entry of certificate revocation request data shall be restricted to Officers and the subject of the certificate to be revoked.
Data Export and Output		The export or output of confidential and security-relevant data shall only be at the request of authorized users
Key Generation	FCS_CKM.1 Cryptographic Key Generation	The capability to request the generation of Component keys (used to protect data in more than a single session or message) shall be restricted to Administrators.
Private Key Load		The capability to request the loading of Component private keys into cryptographic modules shall be restricted to Administrators.
Private Key Storage		The capability to request the decryption of certificate subject private keys shall be restricted to Officers. The TSF shall no provide a capability to decrypt certificate subject private keys that may be used to generate digital signatures. At least two Officers or one Officer and an Administrator, Auditor, or Operator shall be required to request the decryption of a certificate subject private key.
Trusted Public Key Entry, Deletion, and Storage		The capability to change (add, revise, delete) the trusted public keys shall be restricted to Administrators.
Secret Key Storage		The capability to request the loading of KTS secret keys into cryptographic modules shall be restricted to Administrators.

Private and Secret Key Destruction		The capability to zeroize KTS plaintext private and secret keys shall be restricted to Administrators, Auditors, Officers, and Operators.
Private and Secret Key Export		The capability to export a component private key shall be restricted to Administrators. The capability to export certificate subject private keys shall be restricted to Officers. The export of a certificate subject private key shall require the authorization of at least two Officers or one Officer and an Administrator, Auditor, or Operator.
Certificate Status Change Approval		Only Officers and the subject of the certificate shall be capable of requesting that a certificate be placed on hold. Only Officers shall be capable of removing a certificate from on hold status. Only Officers shall be capable of approving the placing of a certificate on hold. Only Officers and the subject of the certificate shall be capable of requesting the revocation of a certificate. Only Officers shall be capable of approving the revocation of a certificate and all information about the revocation of a certificate.

Table 5-2. Access Controls

FDP_ACF.1.3

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

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the [assignment: *none*].

FDP_ITT – Internal TOE transfer

This family provides requirements that address protection of user data when it is transferred between parts of a TOE across an internal channel. This may be contrasted with the FDP_UCT and FDP_UIT families, which provide protection for user data when it is transferred between distinct TSFs across an external channel, and

FDP_ETC and FDP_ITC, which address transfer of data to or from outside the TSF's control.

FDP_ITT.1 Basic internal transfer protection (iteration 3)

This component requires that user data be protected when transmitted between parts of the TOE.

FDP_ITT.1.1

The TSF shall enforce the [*CIMC TOE Access Control Policy specified in chapter 2 of the SPM*] to prevent the [*modification*] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.1 Basic internal transfer protection (iteration 4)

This component requires that user data be protected when transmitted between parts of the TOE.

FDP_ITT.1.1

The TSF shall enforce the [*CIMC TOE Access Control Policy specified in chapter 2 of the SPM*] to prevent the [*disclosure*] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_UCT – Inter-TSF user data confidentiality transfer protection

This family defines the requirements for ensuring the confidentiality of user data when it is transferred using an external channel between distinct TOEs or users on distinct TOEs.

FDP_UCT.1 Basic data exchange confidentiality (iteration 2)

In this component, the goal is to provide protection from disclosure of user data while in transit.

FDP_UCT.1.1

The TSF shall enforce the [*CIMC TOE Access Control Policy specified in chapter 2 of the SPM*] to be able to [*transmit*] objects in a manner protected from unauthorised disclosure.

5.1.1.5 FPT – Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP (User data protection) class; they may even be implemented using the same mechanisms. However, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

FPT_RVM – Reference mediation

The requirements of this family address the "always invoked" aspect of a traditional reference monitor. The goal of this family is to ensure, with respect to a given SFP, that all actions requiring policy enforcement are validated by the TSF against the SFP. If the portion of the TSF that enforces the SFP also meets the requirements of appropriate components from FPT_SEP (Domain separation) and ADV_INT (TSF internals), then that portion of the TSF provides a "reference monitor" for that SFP.

A TSF that implements a SFP provides effective protection against unauthorized operation if and only if all enforceable actions (e.g. accesses to objects) requested by untrusted subjects with respect to any or all of that SFP are validated by the TSF before succeeding. If an action that could be enforceable by the TSF, is incorrectly enforced or incorrectly bypassed, the overall enforcement of the SFP could be compromised. Subjects could then bypass the SFP in a variety of unauthorised ways (e.g. circumvent access checks for some subjects or objects, bypass checks for objects whose protection was assumed by applications, retain access rights beyond their intended lifetime, bypass auditing of audited actions, or bypass authentication). Note that some subjects, the so called "trusted subjects" with respect to a specific SFP, might be trusted to enforce the SFP by themselves, an bypass the mediation of the SFP.

FPT_RVM.1 Non-bypassability of the TSP (iteration 2)

This component requires non-bypassability for all SFPs in the TSP.

FPT_RVM.1.1

The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_ITC – Confidentiality of exported TSF data

This family defines the rules for the protection from unauthorised disclosure of TSF data during transmission between the TSF and a remote trusted IT product. This data could, for example, be TSF critical data such as passwords, keys, audit data, or TSF executable code.

FPT_ITC.1 Inter-TSF confidentiality during transmission (iteration 2)

This component requires that the TSF ensure that data transmitted between the TSF and a remote trusted IT product is protected from disclosure while in transit.

FPT_ITC.1.1

The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.

FPT_ITT – Internal TOE TSF data transfer

This family provides requirements that address protection of TSF data when it is transferred between separate parts of a TOE across an internal channel.

FPT_ITT.1 Basic internal TSF data transfer protection (iteration 3)

This component requires that TSF data be protected when transmitted between separate parts of the TOE.

FPT_ITT.1.1

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

FPT_ITT.1 Basic internal TSF data transfer protection (iteration 4)

This component requires that TSF data be protected when transmitted between separate parts of the TOE.

FPT_ITT.1.1

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

5.1.1.6 FIA – Identification and Authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity.

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

The unambiguous identification of authorized users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies. The families in this class deal with determining and verifying the identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorized user. Other classes of requirements (e.g. User Data Protection, Security Audit) are dependent upon correct identification and authentication of users in order to be effective.

FIA_UAU – User Authentication

This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

FIA_UAU.1 Timing of authentication (iteration 2)

This component allows a user to perform certain actions prior to the authentication of the user's identity.

FIA_UAU.1.1

The TSF shall allow [assignment: *indicate the authentication mode, introduce the authentication data, cancel the login procedure*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

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

FIA_UID – User Identification

This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which require user identification.

FIA_UID.1 Timing of identification (iteration 2)

This component allows users to perform certain actions before being identified by the TSF.

FIA_UID.1.1

The TSF shall allow [assignment: *indicate the identification mode, introduce the identification data, cancel the login procedure*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

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

FIA_USB – User-subject binding

An authenticated user, in order to use the TOE, typically activates a subject. The user's security attributes are associated (totally or partially) with this subject. This family defines requirements to create and maintain the association of the user's security attributes to a subject acting on the user's behalf.

FIA_USB.1 User-subject binding (iteration 2)

This component requires the maintenance of an association between the user's security attributes and a subject acting on the user's behalf.

FIA_USB.1.1

The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

5.1.2 TOE Extended Security Functional Requirements

This class specifies functional requirements for the KTS TOE. These extended functional requirements are extracted from the [CIMC] document.

5.1.2.1 CIMC Extended Security Functional Requirements

FPT – Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP (User data protection) class; they may even be implemented using the same mechanisms. However, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

FPT_CIMC_TSP.1 Audit log signing event

FPT_CIMC_TSP.1.1

The TSF shall periodically create an audit log signing event in which it computes a digital signature, keyed hash, or authentication code over the entries in the audit log.

FPT_CIMC_TSP.1.2

The digital signature, keyed hash, or authentication code shall be computed over, at least, every entry that has been added to the audit log since the previous audit log signing event and the digital signature, keyed hash, or authentication code from the previous audit log signed event.

FPT_CIMC_TSP.1.3

The specified frequency at which the audit log signing event occurs shall be configurable.

FPT_CIMC_TSP.1.4

The digital signature, keyed hash, or authentication code from the audit log signing event shall be included in the audit log.

FDP – User data protection

This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data.

FDP_ACF_CIMC.2 User private key confidentiality protection

Private keys may be used by the KTS for many different purposes and stored for long periods. KTS may store Component keys, KTS personnel keys, and, for key recovery purposes, certificate subject private keys.

FDP_ACF_CIMC.2.1

KTS personnel private keys shall be stored in a FIPS 140-1 validated cryptographic module or stored in encrypted form. If KTS personnel private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-1 validated cryptographic module.

FDP_ACF_CIMC.2.2

If certificate subject private keys are stored in the TOE, they shall be encrypted using a Long Term Private Key Protection Key. The encryption shall be performed by the FIPS 140-1 validated cryptographic module.

FDP_ACF_CIMC.3 User secret key confidentiality protection

Secret (symmetric) keys may be used for several purposes in the KTS. They may be used to encrypt other secret or private keys when they are stored within or exported from the KTS. They may also be used to authenticate subscribers (users) and KTSs. Secret keys must be protected against unauthorized modification and disclosure.

Applicants for certificates may be given PIN or password authenticators. The process for generating and delivering these authenticators to applicants is outside the scope of this document.

FDP_ACF_CIMC.3.1

User secret keys stored within the KTS, but not within a FIPS 140-1 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-1 validated cryptographic module.

FDP_SDI_CIMC.3 Stored public key integrity monitoring and action

These security requirements are designed to detect the unauthorized modification of public keys stored in the KTS.

FDP_SDI_CIMC.3.1

Public keys stored within the KTS, but not within a FIPS 140-1 validated cryptographic module, shall be protected against undetected modification through the use of digital signatures, keyed hashes, or authentication codes.

FDP_SDI_CIMC.3.2

The digital signature, keyed hash, or authentication code used to protect a public key shall be verified upon each access to the key. If verification fails, the TSF shall [assignment: *generate a report error and forbid the use of the public key*].

FDP_ETC_CIMC.5 Extended user private and secret key export

Keys may be exported form cryptographic modules for a variety of reasons, including key backup, replication, and transmission or user private keys generated in the KTS.

FDP_ETC_CIMC.5.1

Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

FDP_CIMC_BKP.1 CIMC backup and recovery

FDP_CIMC_BKP.1.1

The TSF shall include a backup function.

FDP_CIMC_BKP.1.2

The TSF shall provide the capability to invoke the backup function on demand.

FDP_CIMC_BKP.1.3

The data stored in the system backup shall be sufficient to recreate the state of the system at the time the backup was created using only:

a) a copy of the same version of the KTS as was used to create the backup data;

b) a stored copy of the backup data;

c) the cryptographic key(s), if any, needed to verify the digital signature, keyed hash, or authentication code protecting the backup; and

d) the cryptographic key(s), if any, needed to decrypt any encrypted critical security parameters.

FDP_CIMC_BKP.1.4

The TSF shall include a recovery function that is able to restore the state of the system from a backup. In restoring the state of the system, the recovery function is only required to create an "equivalent" system state in which information about all relevant KTS transactions has been maintained.

FDP_CIMC_BKP.2 Extended CIMC backup and recovery

FDP_CIMC_BKP.2.1

The backup data shall be protected against modification through the use of digital signatures, keyed hashes, or authentication codes.

FDP_CIMC_BKP.2.2

Critical security parameters and other confidential information shall be stored in encrypted form only.

FDP_CIMC_CSE.1 Certificate status export

The KTS must be capable of exporting certificate status information. Any message sent by the KTS containing certificate status information must meet the requirements for Certificate Status Export in addition to the requirements for Data Export specified in the FCO and FPT class.

The following requirements apply to Certificate Status Export.

FDP_CIMC_CSE.1.1

Certificate status information shall be exported from the TOE in messages whose format complies with [assignment: *the X.509 standard for CRLs, the OCSP standard as defined by RFC 2560*].

FDP_CIMC_CER.1 Certificate Generation

The functions in this section address the validation, approval, and signing of public key certificates. X.509 public key certificates issued by the KTS must be compliant with the

X.509 standard. Any fields or extensions to be included in an X.509 certificate will either be created by the KTS according to the rules of the X.509 standard or validated by the KTS to ensure compliance.

The data entered in each field and extension to be included in a certificate must be approved. Generally, a certificate field or extension value may be approved in one of four ways:

1 The data may be approved manually by an Officer.

2 An automated process may be used to review and approve the data.

3 The value for a field or extension may be automatically generated by the KTS.

4 The value for a field or extension may be taken from the certificate profile.

FDP_CIMC_CER.1.1

The TSF shall only generate certificates whose format complies with [assignment: *the X.509 standard for public key certificates*].

FDP_CIMC_CER.1.2

The TSF shall only generate certificates that are consistent with the currently defined certificate profile.

FDP_CIMC_CER.1.3

The TSF shall verify that the prospective certificate subject possesses the private key that corresponds to the public key in the certificate request before issuing a certificate, unless the public/private key pair was generated by the TSF, whenever the private key may be used to generate digital signatures.

FDP_CIMC_CER.1.4

If the TSF generates X.509 public key certificates, it shall only generate certificates that comply with requirements for certificates as specified in ITU-T Recommendation X.509. At a minimum, the TSF shall ensure that:

a) The version field shall contain the integer 0, 1, or 2.

b) If the certificate contains an `issuerUniqueID` or `subjectUniqueID` then the `version` field shall contain the integer `1` or `2`.

c) If the certificate contains `extensions` then the `version` field shall contain the integer `2`.

d) The `serialNumber` shall be unique with respect to the issuing Certification Authority.

e) The `validity` field shall specify a `notBefore` value that does not precede the current time and a `notAfter` value that does not precede the value specified in `notBefore`.

f) If the issuer field contains a null `Name` (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical `issuerAltName` extension.

g) If the `subject` field contains a null `Name` (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical `subjectAltName` extension.

h) The `signature` field and the `algorithm` in the `subjectPublicKeyInfo` field shall contain the OID of a FIPS-approved or recommended algorithm.

FDP_CIMC_CRL.1 Certificate revocation list validation

The functions in these requirements address the validation and approval of certificate revocation information.

Certificate revocation lists (CRLs) issued by the KTS shall be compliant with the X.509 standard. Any fields or extensions to be included in a CRL shall be created by the KTS according to the X.509 standard.

FDP_CIMC_CRL.1.1

A TSF that issues CRLs shall verify that all mandatory fields in any CRL issued contain values in accordance with ITU-T Recommendation X.509. At a minimum, the following items shall be validated:

1 If the `version` field is present, then it shall contain a `1`.

2 If the CRL contains any critical extensions, then the `version` field shall be present and contain the integer `1`.

3 If the `issuer` field contains a null `Name` (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical `issuerAltName` extension.

4 The `signature` and `signatureAlgorithm` fields shall contain the OID for a FIPS-approved digital signature algorithm.

5 The `thisUpdate` field shall indicate the issue date of the CRL.

6 The time specified in the `nextUpdate` field (if populated) shall not precede the time specified in the `thisUpdate` field.

FDP_CIMC_OCSP.1 OCSP basic response validation

The functions in these requirements address the validation and approval of certificate revocation information.

OCSP basic responses issued by the KTS shall be compliant with IETF RFC 2560. Any fields or extensions to be included in an OCSP response shall be created by the KTS according to IETF RFC 2560.

FDP_CIMC_OCSP.1.1

If a TSF is configured to allow OCSP responses of the basic response type, the TSF shall verify that all mandatory fields in the OCSP basic response contain values in accordance with IETF RFC 2560. At a minimum, the following items shall be validated:

1 The `version` field shall contain a `0`.

2 If the `issuer` field contains a null `Name` (e.g., a sequence of zero relative distinguished names), then the response shall contain a critical `issuerAltName` extension.

3 The `signatureAlgorithm` field shall contain the OID for a FIPS-approved digital signature algorithm.

4 The `thisUpdate` field shall indicate the time at which the status being indicated is known to be correct.

5 The `producedAt` field shall indicate the time at which the OCSP responder signed the response.

6 The time specified in the `nextUpdate` field (if populated) shall not precede the time specified in the `thisUpdate` field.

5.1.2.1.1 FCO – Communication

This class provides two families specifically concerned with assuring the identity of a party participating in a data exchange. These families are related to assuring the identity of the originator of transmitted information (proof of origin) and assuring the identity of the recipient of transmitted information (proof of receipt). These families ensure that an originator cannot deny having sent the message, nor can the recipient deny having received it.

This section covers cases in which data is to be associated with a user who is not acting locally. In most cases, this will involve data that has been received in a message that has been signed or that contains an authentication code or keyed hash allowing the source of the message to be determined (in which case the data may be associated with the source of the message). Data received over a secure communication channel (e.g., SSL) could be treated similarly.

The security requirements of remote data entry apply whenever data has been received from a remote source that is considered reliable (i.e., the source of the information can be determined). These requirements also apply to communications between physically distributed parts of a single TOE over an untrusted network.

This section also specifies security requirements associated with the export of data from TOEs. The data may be distributed to a device that is outside the boundary of a TOE (either locally or remotely). The remote device or computer may not be directly connected to the TOE. Data export also applies when data is sent between physically distributed subcomponents of a TOE (e.g., data sent between a CA and RA) and the data is transmitted over an untrusted network.

FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin

FCO_NRO_CIMC.3.1

The TSF shall enforce the generation of evidence of origin for certificate status information and all other security-relevant information at all times.

FCO_NRO_CIMC.3.2

The TSF shall be able to relate the identity and [assignment: *originator certificate*] of the originator of the information, and the security-relevant portions of the information to which the evidence applies.

FCO_NRO_CIMC.3.3

The TSF shall verify the evidence of origin of information for all security-relevant information.

FCO_NRO_CIMC.4 Advanced verification of origin

FCO_NRO_CIMC.4.1

The TSF shall, for initial certificate registration messages sent by the certificate subject, only accept messages protected using an authentication code, keyed hash, or digital signature algorithm.

FCO_NRO_CIMC.4.2

The TSF shall, for all other security-relevant information, only accept the information if it was signed using a digital signature algorithm.

5.1.2.1.2 FMT – Security management

This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

FMT_MTD_CIMC.4 TSF private key confidentiality protection

FMT_MTD_CIMC.4.1

KTS private keys shall be stored in a FIPS 140-1 validated cryptographic module or stored in encrypted form. If KTS private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-1 validated cryptographic module.

FMT_MTD_CIMC.5 TSF secret key confidentiality protection

Secret (symmetric) keys may be used for several purposes in the KTS. They may be used to encrypt other secret or private keys when they are stored within or exported from the KTS. They may also be used to authenticate subscribers (users) and Secret keys must be protected against unauthorized modification and disclosure.

Applicants for certificates may be given PIN or password authenticators. The process for generating and delivering these authenticators to applicants is outside the scope of this document.

FMT_MTD_CIMC.5.1

TSF secret keys stored within the TOE, but not within a FIPS 140-1 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-1 validated cryptographic module.

FMT_MTD_CIMC.7 Extended TSF private and secret key export

Keys may be exported form cryptographic modules for a variety of reasons, including key backup, replication, and transmission of user private keys generated in the KTS.

FMT_MTD_CIMC.7.1

Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

FMT_MOF_CIMC.3 Extended certificate profile management

A certificate profile defines the set of acceptable values for fields and extensions in a certificate. Examples of information that may be specified in a certificate profile include:

- Constraints on the key owner's identifier (e.g., subject and/or `subjectAltName` in X.509);
- The set of allowable algorithms for the subject's public/private key pair;
- The certificate issuer's identifier (e.g., issuer and/or `issuerAltName` in X.509);
- The limitations on the length of time for which the certificate is valid;
- Additional information that may/must be included in a certificate (e.g., which extensions may/must be included in an X.509 certificate);
- Whether the subject of the certificate may be a CA;
- The types of operations that may be performed using the private key corresponding to the public key in the certificate (e.g., possible values for `keyUsage` and/or `extKeyUsage` in X.509);
- The policy (policies) under which the certificate may/must be issued.

FMT_MOF_CIMC.3.1

The TSF shall implement a certificate profile and shall ensure that issued certificates are consistent with that profile.

FMT_MOF_CIMC.3.2

The TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- The key owner's identifier;
- The algorithm identifier for the subject's public/private key pair;
- The identifier of the certificate issuer;
- The length of time for which the certificate is valid;

FMT_MOF_CIMC.3.3

If the certificates generated are X.509 public key certificates, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- KeyUsage;
- BasicConstraints;
- CertificatePolicies

FMT_MOF_CIMC.3.4

The Administrator shall specify the acceptable set of certificate extensions.

FMT_MOF_CIMC.5 Extended certificate revocation list profile management

A certificate revocation list profile is used to define the set of acceptable values for fields and extensions in a CRL. Examples of values that may be covered by a certificate revocation list profile include:

- Extensions – the set of extensions that may/must be included in a CRL and the value of each extension's criticality bit.
- Issuer, issuerAltName – the name of the CRL issuer.
- NextUpdate – the lifetime of a CRL.

FMT_MOF_CIMC.5.1

If the TSF issues CRLs, the TSF must implement a certificate revocation list profile and ensure that issued CRLs are consistent with the certificate revocation list profile.

FMT_MOF_CIMC.5.2

If the TSF issues CRLs, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- Issuer;
- IssuerAltName ;
- NextUpdate (i.e., lifetime of a CRL).

FMT_MOF_CIMC.5.3

If the TSF issues CRLs, the Administrator shall specify the acceptable set of CRL and CRL entry extensions.

FMT_MOF_CIMC.6 OCSP profile management

An online certificate status protocol profile is used to define the set of acceptable values for the fields in an OCSP response. The OCSP profile may specify the type(s) of responses that the KTS may generate (i.e., acceptable values for `responseType`) as well as the set of acceptable values for the fields within the acceptable response types. An example of a value that may be covered by an OCSP profile for the basic response type is `ResponderID`, the identifier of the OCSP responder.

FMT_MOF_CIMC.6.1

If the TSF issues OCSP responses, the TSF shall implement an OCSP profile and ensure that issued OCSP responses are consistent with the OCSP profile.

FMT_MOF_CIMC.6.2

If the TSF issues OCSP responses, the TSF shall require the Administrator to specify the set of acceptable values for the responseType field (unless the KTS can only issue responses of the basic response type).

FMT_MOF_CIMC.6.3

If the TSF is configured to allow OCSP responses of the basic response type, the TSF shall require the Administrator to specify the set of acceptable values for the `ResponderID` field within the basic response type.

5.1.2.1.3 FCS – Cryptographic support

The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation.

This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

FCS_CKM_CIMC.5 CIMC private and secret key zeroization

These security requirements specify requirements for the zeroization/destruction of plaintext private and secret keys stored within the KTS.

FCS_CKM_CIMC.5.1

The TSF shall provide the capability to zeroize plaintext secret and private keys within the FIPS 140-1 validated cryptographic module.

5.1.3 TOE Security Assurance Requirements

The assurance components chosen are those specified to comply with assurance level EAL4, as indicated in the following table:

Assurance Class	**Assurance Component**
Configuration Management	ACM_AUT.1, ACM_CAP.4, ACM_SCP.2
Delivery and Operation	ADO_DEL.2, ADO_IGS.1
Development	ADV_FSP.2, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, ADV_RCR.1, ADV_SPM.1
Guidance Documents	AGD_ADM.1, AGD_USR.1
Life Cycle Support	ALC_DVS.1, ALC_LCD.1, ALC_FLR.2, ALC_TAT.1
Tests	ATE_COV.2, ATE_FUN.1, ATE_IND.2, ATE_DPT.1
Vulnerability Assessment	AVA_SOF.1, AVA_VLA.2, AVA_MSU.2

Table 5-3. TOE Security Assurance Requirements

5.1.3.1 ACM – Configuration Management

ACM_CAP - CM capabilities

The capabilities of the CM system address the likelihood that accidental or unauthorised modifications of the configuration items will occur. The CM system should ensure the integrity of the TOE from the early design stages through all subsequent maintenance efforts.

ACM_CAP.4 Generation support and acceptance procedures

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labelling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE.

Providing controls to ensure that unauthorised modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE.

The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorised.

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_AUT - CM automation

The objective of introducing automated CM tools is to increase the effectiveness of the CM system. While both automated and manual CM systems can be bypassed, ignored, or prove insufficient to prevent unauthorised modification, automated systems are less susceptible to human error or negligence.

ACM_AUT.1 Partial CM automation

In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorised. It is the objective of this component to ensure that the implementation representation is controlled through automated means.

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

ACM_SCP - CM scope

The objective of this family is to require items to be included as configuration items and hence placed under the CM requirements of CM capabilities (ACM_CAP). Applying configuration management to these additional items provides additional assurance that the integrity of TOE is maintained.

ACM_SCP.2 Problem tracking CM coverage

A CM system can control changes only to those items that have been placed under CM (i.e., the configuration items identified in the configuration item list). Placing the

TOE implementation and the evaluation evidence required by the other assurance components in the ST under CM provides assurance that they have been modified in a controlled manner with proper authorisations.

Placing security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution.

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

5.1.3.2 ADO – Delivery and Operation

ADO_DEL - Delivery

The requirements for delivery call for system control and distribution facilities and procedures that detail the measures necessary to provide assurance that the security of the TOE is maintained during distribution of the TOE. For a valid distribution of the TOE, the procedures used for the distribution of the TOE address the threats identified in the PP/ST relating to the security of the TOE during delivery.

ADO_DEL.2 Detection of modification

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C 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.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

ADO_IGS – Installation, generation and start-up

Installation, generation, and start-up procedures are useful for ensuring that the TOE has been installed, generated, and started up in a secure manner as intended by the developer. The requirements for installation, generation and start-up call for a secure transition from the TOE's implementation representation being under configuration control to its initial operation in the user environment.

ADO_IGS.1 Installation, generation and start-up procedures

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.

5.1.3.3 ADV – Development

ADV_FSP – Functional Specification

The functional specification is a high-level description of the user-visible interface and behaviour of the TSF. It is an instantiation of the TOE security functional requirements. The functional specification has to show that all the TOE security functional requirements are addressed.

ADV_FSP.2 Fully defined external interfaces

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

ADV_HLD – High-level design

The high-level design of a TOE provides a description of the TSF in terms of major structural units (i.e. subsystems) and relates these units to the functions that they provide. The high-level design requirements are intended to provide assurance that the TOE provides architecture appropriate to implement the TOE security functional requirements.

The high-level design refines the functional specification into subsystems. For each subsystem of the TSF, the high-level design describes its purpose and function, and identifies the security functions contained in the subsystem. The interrelationships of all subsystems are also defined in the high-level design. These interrelationships will be represented as external interfaces for data flow, control flow, etc., as appropriate.

ADV_HLD.2 Security enforcing high-level design

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C 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.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP enforcing and other subsystems.

ADV_IMP – Implementation representation

The description of the implementation representation in the form of source code, firmware, hardware drawings, etc. captures the detailed internal workings of the TSF in support of analysis.

ADV_IMP.1 Subset of the implementation of the TSF

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

ADV_LLD – Low-level design

The low-level design of a TOE provides a description of the internal workings of the TSF in terms of modules and their interrelationships and dependencies. The low-level design provides assurance that the TSF subsystems have been correctly and effectively refined.

For each module of the TSF, the low-level design describes its purpose, function, interfaces, dependencies, and the implementation of any TSP enforcing functions.

ADV_LLD.1 Descriptive low level design

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP enforcing and other modules.

ADV_RCR – Representation correspondence

The correspondence between the various TSF representations (i.e. TOE summary specification, functional specification, high-level design, low-level design, implementation representation) addresses the correct and complete instantiation of the requirements to the least abstract TSF representation provided. This conclusion is achieved by step-wise refinement and the cumulative results of correspondence determinations between all adjacent abstractions of representation.

ADV_RCR.1 Informal correspondence demonstration

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_SPM – Security policy modeling

It is the objective of this family to provide additional assurance that the security functions in the functional specification enforce the policies in the TSP. This is

accomplished via the development of a security policy model that is based on a subset of the policies of the TSP, and establishing a correspondence between the functional specification, the security policy model, and these policies of the TSP.

ADV_SPM.1 Informal TOE security policy model

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modelled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modelled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

5.1.3.4 AGD – Guidance documents

AGD_ADM – Administrator guidance

Administrator guidance refers to written material that is intended to be used by those persons responsible for configuring, maintaining, and administering the TOE in a correct manner for maximum security. Because the secure operation of the TOE is dependent upon the correct performance of the TSF, persons responsible for performing these functions are trusted by the TSF.

Administrator guidance is intended to help administrators understand the security functions provided by the TOE, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information.

AGD_ADM.1 Administrator guidance

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 behaviour 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_USR – User guidance

User guidance refers to material that is intended to be used by nonadministrative human users of the TOE, and by others (e.g. programmers) using the TOE's external interfaces. User guidance describes the security functions provided by the TSF and provides instructions and guidelines, including warnings, for its secure use.

The user guidance provides a basis for assumptions about the use of the TOE and a measure of confidence that non-malicious users, application providers and others exercising the external interfaces of the TOE will understand the secure operation of the TOE and will use it as intended.

AGD_USR.1 User guidance

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 behaviour 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.

5.1.3.5 ATE – Tests

ATE_COV – Coverage

This family addresses those aspects of testing that deal with completeness of test coverage. That is, it addresses the extent to which the TSF is tested, and whether or

not the testing is sufficiently extensive to demonstrate that the TSF operates as specified.

ATE_COV.2 Analysis of coverage

In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an examination of developer analysis of correspondence.

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C 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.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_FUN – Functional tests

Functional testing performed by the developer establishes that the TSF exhibits the properties necessary to satisfy the functional requirements of its PP/ST. Such functional testing provides assurance that the TSF satisfies at least the security functional requirements, although it cannot establish that the TSF does no more than what was specified. The family "Functional tests" is focused on the type and amount of documentation or support tools required, and what is to be demonstrated through developer testing. Functional testing is not limited to positive confirmation that the required security functions are provided, but may also include negative testing to check for the absence of particular undesired behaviour (often based on the inversion of functional requirements).

This family contributes to providing assurance that the likelihood of undiscovered flaws is relatively small.

The families Coverage (ATE_COV), Depth (ATE_DPT) and Functional tests (ATE_FUN) are used in combination to define the evidence of testing to be supplied by a developer. Independent functional testing by the evaluator is specified by Independent testing (ATE_IND).

ATE_FUN.1 Functional testing

The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

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_IND – Independent testing

One objective is to demonstrate that the security functions perform as specified.

An additional objective is to counter the risk of an incorrect assessment of the test outcomes on the part of the developer that results in the incorrect implementation of the specifications, or overlooks code that is non-compliant with the specifications.

ATE_IND.2 Independent testing - sample

The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

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_DPT – Depth

The components in this family deal with the level of detail to which the TSF is tested. Testing of security functions is based upon increasing depth of information derived from analysis of the representations.

The objective is to counter the risk of missing an error in the development of the TOE. Additionally, the components of this family, especially as testing is more concerned with the internal structure of the TSF, are more likely to discover any malicious code that has been inserted.

Testing that exercises specific internal interfaces can provide assurance not only that the TSF exhibits the desired external security behaviour, but also that this behaviour stems from correctly operating internal mechanisms.

ATE_DPT.1 Testing: high level design

The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realised.

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

5.1.3.6 ALC – Life Cycle Support

ALC_DVS – Development security

Development security is concerned with physical, procedural, personnel, and other security measures that may be used in the development environment to protect the TOE. It includes the physical security of the development location and any procedures used to select development staff.

ALC_DVS.1 Identification of security measures

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_LCD – Life cycle definition

Poorly controlled development and maintenance of the TOE can result in a flawed implementation of a TOE (or a TOE that does not meet all of its security requirements). This, in turn, results in security violations. Therefore, it is important that a model for the development and maintenance of a TOE be established as early as possible in the TOE's life-cycle.

Using a model for the development and maintenance of a TOE does not guarantee that the TOE will be free of flaws, nor does it guarantee that the TOE will meet all of its security functional requirements. It is possible that the model chosen will be insufficient or inadequate and therefore no benefits in the quality of the TOE can be observed. Using a life-cycle model that has been approved by some group of experts (e.g. academic experts, standards bodies) improves the chances that the development and maintenance models will contribute to the overall quality of the TOE.

ALC_LCD.1 Developer defined life cycle model

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_FLR – Flaw remediation

Flaw remediation requires that discovered security flaws be tracked and corrected by the developer. Although future compliance with flaw remediation procedures cannot be determined at the time of the TOE valuation, it is possible to evaluate the policies and procedures that a developer has in place to track and correct flaws, and to distribute the flaw information and corrections.

ALC_FLR.2 Flaw reporting procedures

In order for the developer to be able to act appropriately upon security flaw reports from TOE users, and to know to whom to send corrective fixes, TOE users need to understand how to submit security flaw reports to the developer. Flaw remediation guidance from the developer to the TOE user ensures that TOE users are aware of this important information.

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 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_TAT – Tools and techniques

Tools and techniques is an aspect of selecting tools that are used to develop, analyse and implement the TOE. It includes requirements to prevent ill-defined, inconsistent or incorrect development tools from being used to develop the TOE. This includes, but is not limited to, programming languages, documentation, implementation standards, and other parts of the TOE such as supporting runtime libraries.

ALC_TAT.1 Well-defined development tools

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

5.1.3.7 AVA – Vulnerability assessment

AVA_MSU – Misuse

Misuse investigates whether the TOE can be configured or used in a manner that is insecure but that an administrator or user of the TOE would reasonably believe to be secure.

The objectives are:

a) to minimise the probability of configuring or installing the TOE in a way that is insecure, without the user or administrator being able to detect it;

b) to minimise the risk of human or other errors in operation that may deactivate, disable, or fail to activate security functions, resulting in an undetected insecure state.

AVA_MSU.2 Validation of analysis

The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1C 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.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

AVA_SOF – Strength of TOE security functions

Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security behaviour can be made using the results of a quantitative or statistical analysis of the security behaviour of these mechanisms and the effort required to overcome them. The qualification is made in the form of a strength of TOE security function claim.

AVA_SOF.1 Strength of TOE security function evaluation

The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

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_VLA – Vulnerability analysis

Vulnerability analysis is an assessment to determine whether vulnerabilities identified, during the evaluation of the construction and anticipated operation of the TOE or by other methods (e.g. by flaw hypotheses), could allow users to violate the TSP.

Vulnerability analysis deals with the threats that a user will be able to discover flaws that will allow unauthorised access to resources (e.g. data), allow the ability to interfere with or alter the TSF, or interfere with the authorised capabilities of other users.

AVA_VLA.2 Independent vulnerability analysis

A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE.

The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.

AVA_VLA.2.1D The developer shall perfor a vulnerability analysis.

AVA_VLA.2.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.2.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3C 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.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

5.2 Security requirements for the IT environment

5.2.1 Security Functional Requirements for the IT environment

This section specifies the security functional requirements that are applicable to the IT environment. All these requirements has been extracted from the [CIMC] Protection Profile, except the FMT_SMF.1.1 requirement (FMT_SMF Specification of Management Functions) that has been included in order to accomplish dependencies between functional requirements.

Some of these requirements has been instantiated by means the use of the operations mechanism offered by the Common Criteria. The following table lists all the security functional requirements for the IT environment, and the type of operation applied to them.

Functional Requirement	*Security Target Operation*
FAU_GEN.1.1 (FAU_GEN.1 iteration 1)	Selection, Assignment, Refinement
FAU_GEN.1.2 (FAU_GEN.1 iteration 1)	Refinement, Assignment
FAU_GEN.2.1 (FAU_GEN.2 iteration 1)	Refinement

FAU_SAR.1.1	Assignment, Refinement
FAU_SAR.1.2	Refinement
FAU_SAR.3.1	Selection, Assignment, Refinement
FAU_SEL.1.1 (FAU_SEL.1 iteration 1)	Selection, Assignment, Refinement
FAU_STG.1.1 (FAU_STG.1 iteration 1)	Refinement
FAU_STG.1.2 (FAU_STG.1 iteration 1)	Selection, Refinement
FAU_STG.4.1 (FAU_STG.4 iteration 1)	Selection, Assignment, Refinement
FPT_STM.1.1 (FPT_STM.1 iteration 1)	Refinement
FPT_SEP.1.1	Refinement
FPT_SEP.1.2	Refinement
FPT_RVM.1.1 (FPT_RVM.1 iteration 1)	Refinement
FPT_ITC.1.1 (FPT_ITC.1 iteration 1)	Refinement
FPT_ITT.1.1 (FPT_ITT.1 iteration 1)	Selection, Refinement
FPT_ITT.1.1 (FPT_ITT.1 iteration 2)	Selection, Refinement
FPT_AMT.1.1	Selection, Refinement
FMT_SMR.2.1	Assignment, Refinement
FMT_SMR.2.2	Refinement
FMT_SMR.2.3	Assignment, Refinement
FMT_MOF.1.1 (FMT_MOF.1 iteration 1)	Selection, Assignment, Refinement
FMT_MSA.1.1	Selection, Assignment, Refinement
FMT_MSA.2.1	Refinement
FMT_MSA.3.1	Selection, Assignment, Refinement
FMT_MSA.3.2	Assignment, Refinement
FMT_MTD.1.1	Assignment, Selection, Refinement
FMT_SMF.1.1	Assignment, Refinement
FDP_ACC.1.1 (FDP_ACC.1 iteration 1)	Assignment, Refinement
FDP_ACF.1.1 (FDP_ACF.1 iteration 1)	Assignment, Refinement
FDP_ACF.1.2 (FDP_ACF.1 iteration 1)	Assignment, Refinement
FDP_ACF.1.3 (FDP_ACF.1 iteration 1)	Assignment, Refinement
FDP_ACF.1.4 (FDP_ACF.1 iteration 1)	Assignment, Refinement
FDP_ITT.1.1 (FDP_ITT.1 iteration 1)	Assignment, Selectionn, Refinement
FDP_ITT.1.1 (FDP_ITT.1 iteration 2)	Assignment, Selectionn, Refinement

FDP_UCT.1.1 (FDP_UCT.1 iteration 1)	Assignment, Selection, Refinement
FIA_ATD.1.1	Assignment, Refinement
FIA_UAU.1.1 (FIA_UAU.1 iteration 1)	Assignment, Refinement
FIA_UAU.1.2 (FIA_UAU.1 iteration 1)	Refinement
FIA_UID.1.1 (FIA_UID.1 iteration 1)	Assignment, Refinement
FIA_UID.1.2 (FIA_UID.1 iteration 1)	Refinement
FIA_USB.1.1 (FIA_USB.1 iteration 1)	Refinement
FIA_AFL.1.1	Refinement, Selection, Assignment
FIA_AFL.1.2	Assignment, Refinement
FTP_TRP.1.1	Selection, Refinement
FTP_TRP.1.2	Selection, Refinement
FTP_TRP.1.3	Assignment, Selection, Refinement
FCS_CKM.1.1	Assignment, Refinement
FCS_CKM.4.1	Assignment, Refinement
FCS_COP.1.1	Assignment, Refinement
FPT_TST_CIMC.2.1	None
FPT_TST_CIMC.2.2	Assignment
FPT_TST_CIMC.3.1	None
FPT_TST_CIMC.3.2	Assignment

Table 5-4. Functional Requirements for the TOE Environment

5.2.1.1 FAU – Security audit

Security auditing involves recognizing, recording, storing and analyzing information related to security relevant activities (i.e. activities controlled by the TSP). The resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

Audit includes a chronological recording of events that occur in a system. The objective is to track what occurs to enable the reconstruction and examination of a sequence of events and/or changes in an event. This is useful in ensuring that the system is operated securely and in providing evidence when a suspected or actual security compromise has occurred. Audit also provides for reconstructing a specific state of a system. The objective in a PKI system is to enable an appropriate authority to determine whether a signature should have been accepted as valid.

The audit will be used to reconstruct important events that were performed by the TOE, such as issuance of a CA certificate, and the user or event (e.g., a signed

certificate request) that caused them. The audit will be used to arbitrate future disputes by establishing the validity of a signature at a particular time.

The audit log records the security-relevant events that were performed by the TOE and the users or events (e.g., a signed certificate request) that caused them. This subsection specifies the security requirements for maintaining and protecting the integrity of the audit logs.

FAU_GEN – Security Audit Data Generation

FAU_GEN.1 Audit Data Generation (iteration 1)

Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

FAU_GEN.1.1

The [*IT environment*] 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 [*minimum*] level of audit; and

c) [

- *Any changes to the audit parameters, e.g., audit frequency, type of event audited. Any attempt to delete the audit log.*
- *Successful and unsuccessful attempts to assume a role.*
- *The maximum authentication attempts is changed.*
- *Maximum authentication attempts unsuccessful authentication attempts occur during user login.*
- *An Administrator unlocks an account that has been locked as a result of unsuccessful authentication attempts.*
- *An Administrator changes the type of authenticator, e.g., from password to biometrics.*
- *Roles and users are added or deleted.*
- *The access control privileges of a user account or a role are modified.]*

FAU_GEN.1.2

The [*IT environment*] 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, [*none*]

[*Additionally, the audit shall not include plaintext private or secret keys or other critical security parameters.*]

FAU_GEN.2 User Identity Association (iteration 1)

The IT environment shall associate auditable events to individual user identities.

FAU_GEN.2.1

The [*IT environment*] shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_SAR – Security Audit Review

FAU_SAR.1 Audit review

Audit review provides the capability to read information from the audit records.

FAU_SAR.1.1

The [*IT environment*] shall provide [assignment: *Auditors*] with the capability to read [*all information*] from the audit records.

FAU_SAR.1.2

The [*IT environment*] shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_SAR.3 Selectable audit review

Audit review provides the capability to read information from the audit records.

FAU_SAR.3.1

The [*IT environment*] shall provide the ability to perform [*searches*] of audit data based on [*the type of event, the user responsible for causing the event and as specified in Table below*].

Section/Function	Search Criteria
Certificate Request Remote and Local Data Entry	Identity of ghe subject of the certificate being requested
Certificate Revocation Request Remote and Local Data Entry	Identity of the subject of the certificate to be revoked

Table 5-5. Audit Search Criteria

FAU_SEL – Security Audit Event Selection

FAU_SEL.1 Selective Audit (iteration 1)

Selective Audit, requires the ability to include or exclude events from the set of audited events based upon attributes to be specified by the PP/ST author.

FAU_SEL.1.1

The [*IT environment*] shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

[selection: *event type*]

i) [assignment: *none*]

FAU_STG – Security Audit Event Storage

FAU_STG.1 Protected audit trail storage (iteration 1)

Protected audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorised deletion and/or modification.

FAU_STG.1.1

The [*IT environment*] shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2

The [*IT environment*] shall be able to [*detect*] unauthorized modifications to the audit records in the audit trail.

FAU_STG.4 Prevention of audit data loss (iteration 1)

FAU_STG.4 Prevention of audit data loss specifies actions in case the audit trail is full.

FAU_STG.4.1

The [*IT environment*] shall [*prevent auditable events]* except those taken by the [*Auditor*, if the audit trail is full.

5.2.1.2 FPT – Protection of the IT environment

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the IT environment (independent of TSP specifics) and to the integrity of IT environment data (independent of the specific contents of the TSP data).

FPT_STM – Time stamps

FPT_STM.1 Reliable time stamps (iteration 1)

This component requires that the IT environment provide reliable time stamps for IT environment functions.

FPT_STM.1.1

The [*IT environment*] shall be able to provide reliable time stamps for its own use.

FPT_SEP – Domain separation

FPT_SEP.1 TSF domain separation

This component provides a distinct protected domain for the IT environment and provides separation between subjects within the TSC.

FPT_SEP.1.1

[*Each operating system in the IT environment*] shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2

[*Each operating system in the IT environment*] shall enforce separation between the security domains of subjects in [*its scope of control*].

FPT_RVM – Reference mediation

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

This component requires non-bypassability for all SFPs in the TSP.

FPT_RVM.1.1

[*Each operating system in the IT environment*] shall ensure that [*its policy*] enforcement functions are invoked and succeed before each function within [*its scope of control*] is allowed to proceed.

FPT_ITC – Confidentiality of exported TSF data

FPT_ITC.1 Inter-TSF confidentiality during transmission (iteration 1)

This component requires that the IT environment ensure that data transmitted between the IT environment and a remote trusted IT product is protected from disclosure while in transit.

FPT_ITC1.1

The [*IT environment*] shall protect [*confidential IT environment*] data transmitted from the [*IT environment*] to a remote trusted IT product from unauthorized disclosure during transmission.

FPT_ITT – Internal TOE TSF data transfer

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

This component requires that IT environment data be protected when transmitted between separate parts of the TOE Environment IT.

FPT_ITT.1.1

The [*IT environment*] shall protect [*security-relevant IT environment*] data from [*modification*] when it is transmitted between separate parts of the [*IT environment*].

FPT_ITT.1 Basic internal TSF data transfer protection (iteration 2)

This component requires that IT environment data be protected when transmitted between separate parts of the TOE Environment IT.

FPT_ITT.1.1

The [*IT environment*] shall protect [*confidential IT environment*] data from [*disclosure*] when it is transmitted between separate parts of the [*IT environment*].

FPT_AMT – Underlying abstract machine test

FPT_AMT.1 Abstract machine test

This component provides for testing of the underlying abstract machine.

FPT_AMT.1.1

The [*IT environment*] shall run a suite of tests [selection: *during initial start-up*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the [*IT environment*].

5.2.1.3 FMT – Security Management

This class is intended to specify the management of several aspects of the IT environment: security attributes, IT environment data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

FMT_SMR – Security management roles

FMT_SMR.2 Restrictions on security roles

This component specifies the roles with respect to security that the IT environment recognises.

FMT_SMR.2.1

The [*IT environment*] shall maintain the roles [*Administrator, Auditor, and Officer*].

FMT_SMR.2.2

The [*IT environment*] shall be able to associate users with roles.

FMT_SMR.2.3

The [*IT environment*] shall ensure that [*a) no identity is authorized to assume both an Administrator and an Officer role; b) no identity is authorized to assume both an Auditor and a Officer role; and c) no identity is authorized to assume both an Administrator and an Auditor role*].

FMT_MOF – Management of functions in TSF

FMT_MOF.1 Management of security functions behavior (iteration 1)

This component allows the authorized users (roles) to manage the behavior of functions in the IT environment that use rules or have specified conditions that may be manageable.

FMT_MOF.1.1

The [*IT environment*] shall restrict the ability to [*modify the behaviour of*] the functions [*list of functions listed in the table below*] to [*the authorised roles as specified in the table below*]

Section/Function	Component	Function/Authorized Role
Security Audit		The capability to configure the audit parameters shall be restricted to Administrators.
Identification and Authentication		The capability to specify or change maximum authentication attempts shall be restricted to Administrators. The capability to change authentication mechanisms shall be restricted to Administrators.
Account Administrators		The capability to create user accounts and roles shall be restricted to Administrators. The capability to assign privileges to those accounts and roles shall be restricted to Administrators.

Table 5-6. Authorized Roles for Management of Security Functions Behavior

FMT_MSA – Management of security attributes

FMT_MSA.1 Management of security attributes

This component allows authorised users (roles) to manage the specified security attributes.

FMT_MSA.1.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to restrict the ability to [*modify*] the security attributes [assignment: *user definitions, roles*] to [*Administrators*].

FMT_MSA.2 Secure security attributes

This component ensures that values assigned to security attributes are valid with respect to the secure state.

FMT_MSA.2.1

The [*IT environment*] shall ensure that only secure values are accepted for security attributes.

FMT_MSA.3 Static attributes initialization

This component ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.

FMT_MSA.3.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to provide [selection: choose one of: *permissive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The [*IT environment*] shall allow the [*Administrator*] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD – Management of TSF data

FMT_MTD.1 Management of TSF data

This component allows authorised users to manage IT environment data.

FMT_MTD.1.1

The [*IT environment*] shall restrict the ability to [*view (read) or delete*] the [*audit logs*] to [*Auditors*].

FMT_SMF – Specification of Management Functions

FMT_SMF.1 Specification of Management Functions

This component requires that the environment provide specific management functions.

FMT_SMF.1.1

The [*IT environment*] shall be capable of performing the following security management functions: [assignment: *management of users and permissions of access on the part of the users, administration of users authentication*]

5.2.1.4 FDP – User Data Protection

This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. FDP is split into four groups of families (listed below) that address user data within a TOE, during import, export, and storage as well as security attributes directly related to user data.

FDP_ACC – Access control policy

FDP_ACC.1 Subset access control (iteration 1)

This component requires that each identified access control SFP be in place for a subset of the possible operations on a subset of the objects in the TOE.

FDP_ACC.1.1

The [*IT environment*] shall enforce the [CIMC IT Environment Access Control Policy *specified in chapter 2of the SPM*] on [assignment: all *users, files and other structures containing sensitive information and all operations among users and objects covered by the CIMC IT Environment Access Control Policy*]

FDP_ACF – Access control functions

FDP_ACF.1 Security attribute based access control (iteration 1)

This component allows the IT environment to enforce access based upon security attributes and named groups of attributes. Furthermore, the IT environment may have the ability to explicitly authorize or deny access to an object based upon security attributes.

FDP_ACF.1.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to objects based on the following: [*the identity of the subject and the set of roles that the subject is authorized to assume*].

FDP_ACF.1.2

The [*IT environment*] shall enforce the following [*rule]* to determine if an operation among controlled subjects and controlled objects is allowed: [*the capability to zeroize plaintext private and secret keys shall be restricted to Administrators, Auditors, Officers, and Operators*].

FDP_ACF.1.3

The [*IT environment*] shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *none*].

FDP_ACF.1.4

The [*IT environment*] shall explicitly deny access of subjects to objects based on the [assignment: *none*].

FDP_ITT – Internal TOE transfer

FDP_ITT.1 Basic internal transfer protection (iteration 1)

This component requires that user data be protected when transmitted between parts of the TOE Environment IT.

FDP_ITT.1.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to prevent the [*modifications of security-relevant*] of user data when it is transmitted between physically-separated parts of the [*IT environment*].

FDP_ITT.1 Basic internal transfer protection (iteration 2)

This component requires that user data be protected when transmitted between parts of the TOE.

FDP_ITT.1.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to prevent the [*disclosure of confidential*] of user data when it is transmitted between physically-separated parts of the [*IT environment*].

FDP_UCT – Inter-TSF user data confidentiality transfer protection

FDP_UCT.1 Basic data exchange confidentiality (iteration 1)

In this component, the goal is to provide protection from disclosure of user data while in transit.

FDP_UCT.1.1

The [*IT environment*] shall enforce the [*CIMC IT Environment Access Control Policy specified in chapter 2 of the SPM*] to be able to [*transmit*] objects in a manner protected from unauthorised disclosure.

5.2.1.5 FIA – Identification and Authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity.

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies. The families in this class deal with determining and verifying the identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorised user. Other classes of requirements (e.g. User Data Protection, Security Audit) are dependent upon correct identification and authentication of users in order to be effective.

FIA_ATD – User attribute definition

FIA_ATD.1 User attribute definition

This component allows user security attributes for each user to be maintained individually.

FIA_ATD.1.1

The [*IT environment*] shall maintain the following list of security attributes belonging to individual users: [*the set of roles that the user is authorized to assume, [assignment: no other security attributes]*].

FIA_UAU – User Authentication

FIA_UAU.1 Timing of authentication (iteration 1)

This component allows a user to perform certain actions prior to the authentication of the user's identity.

FIA_UAU.1.1

The [*IT environment*] shall allow [assignment: *request for username and password*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

The [*IT environment*] shall require each user to be successfully authenticated before allowing any other [*IT environment*] -mediated actions on behalf of that user.

FIA_UID – User Identification

FIA_UID.1 Timing of identification (iteration 1)

This component allows users to perform certain actions before being identified by the IT environment.

FIA_UID.1.1

The [*IT environment*] shall allow [assignment: *request for username and password*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

The [*IT environment*] shall require each user to be successfully identified before allowing any other [*IT environment*] -mediated actions on behalf of that user.

FIA_USB – User-subject binding

FIA_USB.1 User-subject binding (iteration 1)

This component requires the maintenance of an association between the user's security attributes and a subject acting on the user's behalf.

FIA_USB.1.1

The [*IT environment*] shall associate the appropriate user security attributes with subjects acting on behalf of that user.

FIA_AFL – Authentication failures

FIA_AFL.1 Authentication failure handling

This component requires that the IT environment be able to terminate the session establishment process after a specified number of unsuccessful user authentication attempts. It also requires that, after termination of the session establishment process, the IT environment be able to disable the user account or the point of entry (e.g. workstation) from which the attempts were made until an administrator-defined condition occurs.

FIA_AFL.1.1

[*If authentication is not performed in a cryptographic module that has been FIPS 140-1 validated to an overall Level of 2 or higher with Level 3 or higher for Roles and Services*], the [*IT environment*] shall detect when an [*Administrator*] [*configurable maximum authentication attempts*] unsuccessful authentication attempts have occurred [*since the last successful authentication for the indicated user identity*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met or surpassed, the [*IT environment*] shall [assignment: *record a log related to the authentication failure*].

5.2.1.6 FTP – Trusted path/channels

Families in this class provide requirements for a trusted communication path between users and the IT environment, and for a trusted communication channel between the IT environment and other trusted IT products.

FTP_TRP – Trusted path

FTP_TRP.1 Trusted path

This component requires that a trusted path between the IT environment and a user be provided for a set of events defined by a PP/ST author. The user and/or the IT environment may have the ability to initiate the trusted path.

FTP_TRP.1.1

The [*IT environment*] shall provide a communication path between itself and [selection: *local, remote*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2

The [*IT environment*] shall permit [selection: *the IT environment, local users, remote users*] to initiate communication via the trusted path.

FTP_TRP.1.3

The [*IT environment*] shall require the use of the trusted path for [*initial user authentication]*, [assignment: *no other services]*

5.2.1.7 FCS – Cryptographic support

The IT environment may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

FCS_CKM – Cryptographic key management

FCS_CKM.1 Cryptographic key generation

This component requires cryptographic keys to be generated in accordance with a specified algorithm and key sizes which can be based on an assigned standard.

FCS_CKM.1.1

The [*FIPS 140-1 validated cryptographic module*] shall generate cryptographic keys in accordance with [assignment: *3DES, DES, AES, RSA, DSA*] that meet the following: [assignment: *FIPS 46-3 Data Encryption Standard (DES, 3DES), FIPS PUB 186-2 (DSA and RSA), FIPS PUB 197 (AES)]*

FCS_CKM.4 Cryptographic key destruction

This component requires cryptographic keys to be destroyed in accordance with a specified destruction method which can be based on an assigned standard.

FCS_CKM.4.1

The [*IT environment*] shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *any FIPS approved or*

recommended key destruction method] that meets the following: [assignment: *FIPS 140-2*]

FCS_COP – Cryptographic operation

FCS_COP.1 Cryptographic operation

This component requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.

FCS_COP.1.1

The [*FIPS 140-1 validated cryptographic module*] shall perform [assignment: *encryption, decryption, signature generation, signature verification, hash generation, hash verification*] in accordance with [assignment: *FIPS 46-3 Data Encryption Standard -DES, 3DES- (encryption, decryption), FIPS PUB 186-2 –DSA, RSA- (signature generation, signature verification), FIPS PUB 197 –AES- (encryption, decryption), FIPS PUB 180-2 - SHA1, SHA-256, SHA-512, SHA-384- (hash generation, hash verification)]*.

5.2.2 Propietary Extended Security Requirements for the IT environment

This section specifies propietary extended security requirements for the IT environment.

5.2.2.1 FPT - Protection of the TSF

This class contains families of requirements that relate to the integrity and management of the mechanisms that provide the TSF and to the integrity of TSF data. This class also contains requirements that are related to access control mechanisms.

FPT_ACC – Access Control

This family defines requirements about the access control to the tools and programs that can be available by the TOE.

FPT_ACC.1 Access Control to the software

This component requires access control measures to be applied to those software that can be available by the TOE.

FPT_ACC.1.1

The environment must not have installed any database program (e.g. telnet, import, export, …) that access to the database used by the TOE.

5.2.3 Propietary Extended Security Non-IT Requirements for the environment

This section specifies propietary extended security non-it requirements for the environment.

5.2.3.1 FPT - Protection of the TSF

This class contains families of requirements that relate to the integrity and management of the mechanisms that provide the TSF and to the integrity of TSF data. This class also contains requirements that are related to access control mechanisms.

FPT_ACC – Access Control

This family defines requirements about the access control to the tools and programs that can be available by the TOE.

FPT_ACC.1 Access Control to the software

This component requires access control measures to be applied to those software that can be available by the TOE.

FPT_ACC.1.2

If programs that access to the database are used, then this access must be controlled and supervised by the Auditor.

5.2.4 CIMC Extended Security Functional Requirements

These extended functional requirements are extracted from the [CEN01c] and [CIMC] documents.

FPT – Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP (User data protection) class; they may even be implemented using the same mechanisms. However, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

FPT_TST_CIMC.2 Software/firmware integrity test

FPT_TST_CIMC.2.1

An error detection code (EDC) or FIPS-approved or recommended authentication technique (e.g., the computation and verification of an authentication code, keyed

hash, or digital signature algorithm) shall be applied to all security-relevant software and firmware residing within the KTS (e.g., within EEPROM and RAM). The EDC shall be at least 16 bits in length.

FPT_TST_CIMC.2.2

The error detection code, authentication code, keyed hash, or digital signature shall be verified at power-up and on-demand. If verification fails, the IT environment shall [assignment: *report the test failure*].

FPT_TST_CIMC.3 Software/firmware load test

FPT_TST_CIMC.3.1

A cryptographic mechanism using a FIPS-approved or recommended authentication technique (e.g., an authentication code, keyed hash, or digital signature algorithm) shall be applied to all security-relevant software and firmware that can be externally loaded into the KTS.

FPT_TST_CIMC.3.2

The IT environment shall verify the authentication code, keyed hash, or digital signature whenever the software or firmware is externally loaded into the KTS. If verification fails, the IT environment shall [assignment: *does not allow the execution of the component where the test has failed*].

CHAPTER 6

6 TOE Summary Specification

6.1 TOE Security Functions

In this section a description of the security functions of the TOE that meet the TOE security requirements is provided.

The explanation of how the security requirements are accomplished by means the security functions is described in this section.

The TOE provides the following Security Function Families:

- Audit Data Management
- Secure Database
- Access Control Management
- Identification and Authentication
- Secure Communications
- Certification Management
- Private Secure Store
- Key Archive Management
- Backup and Recovery

6.1.1 Audit Data Management

KeyOne 3.0 keeps information on the operations performed by maintaining an event log. Recorded operations include those done by administrators or other users using the KeyOne applications, but also operations executed internally by some online servers installed with the product. Some examples of logged operations are the approval of a certificate request, the revocation of a certificate, the processing of a batch, the generation of CRLs. By means a configuration option of the KeyOne Console application is possible to configure the list of events to register, so that the events can be included or excluded of the list of events generated by the KeyOne system.

Operations are divided into events, so that information on one or more events is stored for each relevant operation (for example, the generation of the various requested certificates when a batch is processed by KeyOne CA). Both informative and error events are logged. Event information is stored in the KeyOne product database, in a separate log table. This table may be configured to reside in a different database than the rest of KeyOne tables, but always it resides in a i3D database, and therefore the event logs have all the security mechanisms provided by the i3D technology (integrity, authentication, non-repudiation).

6.1.1.1 Functional Requirements satisfied by Security Functions

The Audit Data Management services are composed of the following security functions:

- Selective Logs Function (FUNC_SELL). This functionality allows configuring the events to audit. The KeyOne Console application have an option by means the administrator can select the types of events to include/exclude from the total list of events that the Security Audit Data Generation Function could to register.
- Security Audit Data Generation Function (FUNC_SADG). This service is in charge of register in the logs table, information about the events that occur in the system.

These services satisfy the following requirements:

6.1.1.1.1 FAU_GEN.1.1 (iteration 2)

The TOE Security Audit Data Generation Function is able to generate an audit record with the following auditable events:

a) Start-up and shutdown of the audit functions. The audit functions are always started/stopped when the keyOne Servers engine starts/shutdowns. It is not possible with KeyOne applications to start only the Security Audit Data Generation Services without to start the KeyOne Server, and it is not possible to shutdown the Security Audit Data Generation Function without to shutdown the KeyOne Server. When the KeyOne server starts, an audit record is generated in the i3D Database Logs Table, and when the KeyOne server shutdowns then also an audit record is generated indicating that the KeyOne application has been stopped.

b) The following auditable events (corresponding to the minimum level of audit of the FAU_GEN.1.1 requirement):

   a) All modifications to the audit configuration that occur while the audit collection functions are operating (FAU_SEL.1 dependency). All the changes related to the audit configuration will be registered in an audit record: modifications of the list of events that must be audited (Selective Logs Function), changes to the configuration parameters of the logs table and the database where the logs table is stored (change of the logs table, change of the connection driver of the database, change of the database service, change of the user and password related to the database).

   b) Regarding to the changes to the time (FPT_STM.1 dependency), the TOE relies in the system clock. The changes to the system clock are out of the

functionality offered by the KeyOne components, and they are out of the KeyOne control. Therefore, the register of the time changes is responsibility of the IT Environment Access Control.

c) Unsuccessful use of the user identification mechanism, including the user identity provided (FIA_UID.1 dependency). The Security Audit Data Generation Function registers all the attempts of access to the KeyOne system; these attempts imply to use the user identification mechanism, and therefore this event is register. The identity provided in the identification attempts are also included in the log registered (depending on the type of identification, the username or the certificate subject).

d) Modifications to the group of users that are part of a role (FMT_SMR.1 dependency). KeyOne application users belong to one or more groups, and they are both defined in the whole KeyOne system. To each group of users one or more roles, which are specific for each application, can be assigned. These roles are part of the KeyOne Console configuration and are initialized from the values defined by the security policy selected during the start-up. All the modifications over the relationship between the groups of users and roles are registered in an audit registry in the logs table.

e) Successful requests to perform an operation on an object covered by the SFP (FDP_ACF.1 dependency). All operations on objects covered by the Security Functional Policy (roles, keys, …) are registered in an audit log. The following events are registered by the Security Audit Data Generation Function:

- Create, delete and modify users
- Suspend and enable users
- Modify user properties
- Create, delete and modify groups
- Modify group properties
- Modify password restrictions
- Modify the list of the system's CA certificates
- Modify incompatibilities among roles
- Modify roles assigned to groups
- Create the logs table
- Rename the logs table
- Modify the connections with the configured databases

f) Unsuccessful use of the authentication mechanism (FIA_UAU.1 dependency). The Security Audit Data Generation Function registers all the attempts of access to the KeyOne system; these attempts imply to use the user authentication mechanism (user password, or challenge-response).

g) Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject) (FIA_USB.1 dependency).All KeyOne Console users hold one or more roles. Theses roles are part of the KeyOne Console configuration and are initialized from the values defined by the security policy selected during the start-up. It is not possible to directly assign roles to individual users, but to groups. This way, users hold roles according to those assigned to the groups they belong to. The relationship between users and groups, and the relationship between groups and roles can be modified between the KeyOne Console configuration. Any attempt to change these relationships is logged.

h) Successful transfers of user data, including identification of the protection method used (FDP_ITT.1 dependency). This event affects to transfers of user data between an internal channel. From the KeyOne point of view, these transfers correspond to the communication between the KeyOne LRA and KeyOne CA, and the communication between the KeyOne CA and KeyOne VA. The transfer protocol used by these communications is the SSL/TLS protocol. A log register is generated when the KeyOne service starts; this register contains the SSL/TLS connection parameters (algorithms, version of the protocol, …) used in each SSL/TLS connection (it is necessary to stop the server in order to change these parameters), and therefore it includes identification of the protection method used.

   In the communication between the KeyOne LRA and the KeyOne CA, the user data are included in a KeyOne batch. This KeyOne batch contains a digital signature of all the data that it includes, and also it contains the algorithms identifiers used in the digital signature generation.

   In the communication between the KeyOne CA and the KeyOne VA, the user data are included in the NDCCP message. This message contains a digital signature of all the data that it includes, and it contains also the algorithms identifiers used in the digital signature generation.

i) The identity of any user or subject using the data exchange mechanisms (FDP_UCT.1 dependency). This event affects to transfers of user data between an external channel. From the KeyOne system point of view, these transfers correspond to the following communications:

   - Communications between a user and the KeyOne VA using the OCSP protocol. If the identity of the requestor is included in the OCSP request, then it will be included in the log registry indicating the arrival of an OCSP request (`requestorName` field of the OCSP request). If the `requestorName` field is not included in the OCSP request (not signed request), then if the OCSP client is using the SSL/TLS protocol (with client authentication) the issuer and serial number fields are registered in the log registry. If the OCSP client does not use the SSL/TLS with client authentication in order to communicate with the OCSP server then the IP client address will be included in the log entry[5].

     When the KeyOne VA server generates the OCSP response and it sends this message to the user, then a log registry is generated indicating the KeyOne OCSP identity (this identification is indicated by means the

[5] *The user identification is registered in the "observation" field of the log table.*

registration of the <server IP address><service name> in the `author` field of the log table).

- Communications between the KeyOne applications and the database. These communications imply the creation of an registry in the database, and therefore the data involved in these communications are registered.
- Communications between the KeyOne applications and the Hardware Security Module. The communications that involve user data are registered in a log entry by the Security Audit Data Generation Function (eg. creation of user certificates).
- Communications between the KeyOne applications and the Signature Device Creation. The communications that involve user data are registered in a log entry by the Security Audit Data Generation Function (eg. creation of user certificates).

j) Success and failure of the key destruction activity (FCS_CKM.4 dependency). When the KeyOne system deletes the application keys (infrastructure and control keys, and keys for generating certificates and CRLs), a log entry is generated in the log table.

k) Use of the management functions (FMT_SMF.1 dependency). The Security Audit Data Generation Function registers events related to the functions invoked by and administrator operating over aspects associated with the TOE security, as attributes that protect data, attributes that protect the TOE, audit attributes, and identification/authentication attributes. The following events are registered by the Security Audit Data Generation Function:

- Create, delete and modify users
- Suspend and enable users
- Modify user properties
- Create, delete and modify groups
- Modify group properties
- Modify password restrictions
- Modify the list of the system's CA certificates
- Select user certificates
- Modify incompatibilities among roles
- Modify roles assigned to groups
- Select the database connection
- Create the logs table
- Rename the logs table
- Modify the connections with the configured databases

- Select the list of events to audit

l) All offered and rejected values for a security attribute (FMT_MSA.2 dependency). When a value is intended for assigning to a security attribute (for instance, operation of assign a password or certificate to a user), then the related log registry will contain this initial value for the attribute; if the value is rejected, then this rejected value is also included in the unsuccessful operation.

m) Success and failure of the cryptographic key generation and cryptographic key distribution activity (FCS_CKM.1, FCS_CKM.2 dependency).

When the own cryptographic keys are generated, a log registry is generated containing information about the key generation event.

Regarding to the cryptographic key distribution activity:

- The generation of a certificate implies the generation of a log registry containing information about the certificate generation event.

c) The following auditable events:

a) Security audit events. The Security Audit Data Generation Function registers the events related to any changes to the audit parameters. The following events will be registered in the logs table:

- Modifications of the list of events that must be audited
- Changes to the configuration parameters of the logs table and the database where the logs table is stored (change of the logs table, change of the connection driver of the database, change of the database service, change of the user and password related to the database).

The TOE application does not have any functionality in order to delete the audit logs; because using the KeyOne application it is not possible no delete these logs, then not log registry is related to this event.

b) All security-relevant data that is entered in the system (Local Data Entry). All operations that receive locally security-relevant data imply the generation of a log entry. The following events are registered by the Security Audit Data Generation Function:

- Create, delete and modify users
- Suspend and enable users
- Modify user properties
- Create, delete and modify groups
- Modify group properties
- Modify password restrictions
- Modify the list of the system's CA certificates
- Select user certificates

- Modify incompatibilities among roles
- Modify roles assigned to groups
- Select the database connection
- Create the logs table
- Rename the logs table
- Modify the connections with the configured databases
- Select the list of events to audit

c) All security-relevant, messages that are received by the system (Remote Data Entry). This event is related to the entry data that are received remotely, and that it is possible to identify and authenticate the sender of the data. In the KeyOne system context, these events are the ones related to the reception of a signed (because authentication is required) request from an OCSP client to the KeyOne VA server. All the information about the OCSP requests/responses received/sent by the KeyOne VA server is included in a log entry. The information registered in the `observation` field of the log table is the following:

- If the OCSP request is signed, then the `requestorName` field is registered. If the OCSP is not signed, and the communication mechanism used is the TLS/SSL protocol with client authentication, then the issuer and serialNumber of the client certificate used in the TLS/SSL protocol is registered in the `observations` field of the log table. If the OCSP request is not signed, and the communication does not use the TLS/SSL protocol with client authentication, then the client IP address is stored in the `observations` field.
- The server identification (<IP server machine><server name>) is registered in the `author` field.
- If the content-type of the response if "`application/ocsp-request`", then the following information is registered: a) the response status is registered; b) if the response status is different that `malformedRequest`, then the certificates identification (position of the certificate if the OCSP request, hash algorithm, hash of the public key, hash of the issuer name, serial number) is registered; c) if the response status is `successful`, then the following information is registered: status of the certificates (status and revocation reason if the status is revoked), and data of the response signature (`producedAt` field of the OCSP response); e) information about the error (if produced).
- If the content-type of the request is not "`application/ocsp-request`", then all the message is registered.

d) All successful and unsuccessful requests for confidential and security-relevant information (Data Export and Output)

- Regarding to the local data exportation, all the requests that imply an exportation of confidential data (e.g. PKCS #12 requests) are logged.

- Regarding to the remote data output, all the remote requests that can imply confidential traffic (e.g. certification requests from the RA to the CA) are logged.

e) All the requests generation of a cryptographic key (the generation of single session or one-time use symmetric keys is not included in this event). The Security Audit Data Generation Function registers all the requests for generation of symmetric and asymmetric keys. In the start-up phase of the system, an initial log entry regarding to the system creation is generated; because this system creation implies the generation of keys, in this case the log regarding to the generation of keys during the system creation is implicit to the system generation log entry.

f) The loading of Component private keys (Private Key Load). Because the Keyone functionality does not allow a means to load component private keys, then no log entry is generated for this event. All the private keys are generated and maintained in cryptographic modules, and these components are outside the TOE and belonging to the IT environment.

g) All access to certificate subject private keys retained within the TOE for key recovery purposes (Private Key Storage). The recovery operation offered by the KeyOne Key Archive component (component located in the KeyOne CA product) is logged by the Security Audit Data Generation Function.

h) The manual entry of secret keys used for authentication (Secret Key Storage). Because the KeyOne applications do not allow the manual entry of secret keys, no log entry related to this event is generated.

i) The export of private and secret keys (keys used for a single session or messages are excluded of this event). The Security Audit Data Generation Function registers the following events:

- Regarding to exportation of the user private keys, the exportation operation of PKCS #12 implies the generation of a log entry.

- Regarding to the exportation of the TSF private/secret keys, they are exported to the Hardware Security Module when the system is started. In this case, when the system starts, a log entry is generated indicating that the system is running. The log regarding to the exportation of private/secret keys is implicit to the log entry generated in the system start phase.

j) All certificate requests (FDP_CIMC_CER.1). All the certificate requests generated from the KeyOne LRA and from the KeyOne CA are registered by the Security Audit Data Generation Function in audit logs.

k) All requests to change the status of a certificate (Certificate Status Change Approval). All the requests to change the status of a certificate from the KeyOne LRA and from the KeyOne CA are registered by the Security Audit Data Generation Function in audit logs.

l) Any security-relevant changes to the configuration of the TSF (CIMC Configuration). All the changes related to the configuration are logged by the Security Audit Data Generation Function. These services generate a log entry for each following event:

- Created, delete and modify users.
- Suspend and enable users.
- Modify user properties.
- Create, delete and modify groups.
- Modify group properties.
- Modify password restrictions.
- Modify the list of the system's CA certificates.
- Select user certificates.
- Modify incompatibilities among roles.
- Modify roles assigned to groups.
- Select the database connection.
- Create the logs table.
- Rename the logs table.
- Modify the connections with the configured databases.
- Select the list of events to audit

m) All changes to the certificate profile (FMT_MOF_CIMC.2, FMT_MOF_CIMC.3). The Security Audit Data Generation Function generates a log entry for each change to the certificate profile.

n) All changes to the revocation profile (Revocation Profile Management). The Security Audit Data Generation Function generates a log entry for each change to the revocation profile.

o) All changes to the certificate revocation list profile (FMT_MOF_CIMC.4, FMT_MOF_CIMC.5). The Security Audit Data Generation Function generates a log entry for each change to the revocation profile.

p) All changes to the OCSP profile (FMT_MOF_CIMC.6). The Security Audit Data Generation Function generates a log entry for each change to the OCSP profile.

6.1.1.1.2 FAU_GEN.1.2 (iteration 2)

The TOE Security Audit Data Generation Function includes in each log entry the following information:

- Date and time when the event occurred. The date/time is represented in numeric (`time_t`) format (`timelog` field).
- Identification of the entity that generated the event (`author` field).
- A character string indicating the type of entity that generated the event (`role` field).

- A number indicating the type of event (`evtype` field).
- A number that uniquely identifies the event among the set of events of the same type and generated by the same module (`event` field).
- A number identifying the module that generated the event (`modu` field). This column contains a null value for events of type MARK.
- A number that indicates the importance of the event (`evlevel` field). Logs are classified in the following categories according to their importance:
  - Informational: events of this category provide information in operations that were successfully performed. This category implies a successful operation.
  - Mark: whenever an administration session is started and finished, an event of this category is recorded. This category implies a successful operation.
  - Warning: indicates that an unusual condition was detected during an operation, but this did not cause the operation fail. This category implies a failure operation.
  - Error indicates that an operation failed due to a predictable error. This category implies a failure operation.
  - Fatal error: indicates that an unpredictable exceptional circumstance occurred during an operation. This category implies a failure operation.
- An string describing the event. For some events, the description is followed by a list of parameters (separated for new-line characters) whose value will vary depending on the data over which the operation was executed (`obser` field).

Additionally, the following information is registered:

- In the audit log signing event: digital signature, keyed hash and authentication code is included in the audit log (FPT_CIMC_TSP.1).

  The session start and end records are asymmetrically signed with the user's digital signature certificate (signature field). Besides, all records of the session table are linked in a way that a possible fraudulent intermediate session insertion or deletion would be detected when verifying the database integrity. This linkage is done in the following way:

  - The asymmetric signature of the session start record includes the signature field value of the previous session start record.
  - The asymmetric signature of the session end record includes the value of the signature field of the corresponding session start record.

  When closing an i3D session, the session end record that was inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session (`hashchain` field) is added to this record. If the session only consisted on query operations then this field will remain empty. Once updated, the session end

record is signed asymmetrically again with the user's digital signature certificate and the session is considered closed.

Additionally, when an historic record is added (insertion, update or deletion of a logical record), a symmetrical signature of this historic record is generated and it is added into the historic table and also into the related browsing record.

- In the events that imply that security-relevant data are entered in the system, the following information must be included in the registry log: the identity of the data entry individual if the entered data is linked to any other data; this is included with the accepted data (Local Data Entry). The Security Audit Data Generation Function includes in the log registry the identity of the entity responsible of the event, the data entered in the system, and the operations performed by the entity related to the event.
- In the events that imply the requests generation of a cryptographic key (not included the single session or one-time use symmetric keys): the public component of the asymmetric key pair generated is included in the log entry (FCS_CKM.1). The Security Audit Data Generation Function includes this component in the following operations:
  - Request of asymmetric key pair generation.
  - Request of PKCS #12 generation.
  - Request of certificate generation.
- In the events that imply changes to the trusted public keys, including additions and deletions: the public key and all information associated with the key is included in the log registry (Trusted Public Key Entry, Deletion and Storage). When any operation involving trusted public keys (root CA certificates) occurs, then a log registry is generated, containing the public key involved in the operation.
- For each certificate request, a log entry is generated containing the following information (FDP_CIMC_CER.1):
  - If the request is accepted, then a copy of the certificate is included in the certificates table. The entry generated in this table is univocally linked with the log entry containing the related request (by means the unique public key contained in both the request and the certificate).
  - If the request is rejected, then a reason for rejection is included in the log entry.
- For each request to change the status of a certificate: information about whether the request was accepted or rejected is included in the log entry (Certificate Status Change Approval).
- For each change to the certificate profile: the changes made to the profile are registered in the log entry (FMT_MOF_CIMC.2, FMT_MOF_CIMC.3).
- For each change to the revocation profile: the changes made to the profile are registered in the log entry (Revocation Profile Management).
- For each change to the certificate revocation list profile: the changes made to the profile are registered in the log entry (FMT_MOF_CIMC.4, FMT_MOF_CIMC.5).

The Security Audit Data Generation Function does not include in the log entry, the plaintext private or secret keys or other critical security parameters.

6.1.1.1.3 FAU_GEN.2.1 (iteration 2)

Because the Security Audit Data Generation Function always registers in the log entry the identification of the entity that generated the event, then always the association between each auditable event and the identity of the user that caused the event is registered.

6.1.1.1.4 FAU_SEL.1.1 (iteration 2)

The Selective Logs Function allows including or excluding auditable events from the set of audited events, based on the event type. This function provides functionality in order to configure the events to register in the log table, from the KeyOne Console application. This application has an option that graphically shows the current events that will be audited; the application allow changing from this option the events to include/exclude from the showed list.

6.1.2 Secure Database

KeyOne system uses i3D databases. The i3D technology has the following properties:

- Allows to verify the database integrity, this is, detect possible fraudulent data manipulation.
- Assure non-repudiation by the authors of operations performed over data. This is accomplished through digital signatures.
- Keep a historic record of data update, this is, it stores successive versions of each record resulting from various operations performed over the record. This allows keeping a record of the operations performed and avoids loosing digital signatures performed previously by other users when updating data.
- Allows concurrent access to the same database tables by several users.
- It works over any SQL database management system. i3D functionality fully resides in the client's system, without the necessary existence of an intermediate server.

Operations are grouped in sessions (i3d sessions), so in order to consult or carry out changes in a table the user must open a session first with that table. After some time, once the desired operations have finished, the user must close the session. Sessions performed by the various users over a certain table are identified by a sequential number called session identifier.

Entities that access an i3D database are classified as:

- Users or entities that perform operations over data. These operations include reading, insertion, update and deletion of database table records. Each user must have a own digital signature certificate that will be used to sign data that the user has added, updated or deleted during the i3D session.

- There are entities that can perform certain special operations over the database (master entities). These entities must have a digital signature certificate and a data encipherment certificate. Functions reserved for entities enabled as masters are:
    - Verify and close i3D sessions that were not closed in an orderly way by the users (for instance, in case of disaster).
    - Sign already close i3D sessions again in order to force the recovery of data integrity.

When starting an administration session, then an i3D session is started with each one of the product tables. The various operations performed by the application users (certificate request approval, certificate revocation, CRL generation, batch processing, …) cause the insertion of new historic records in the i3D tables, so that the database internally keeps the successive updates of each logical record. When the contents of a table are consulted from the administration application, the last version of the records is always shown.

I3D sessions are automatically closed when ending an administration session; this is, when the server is closed by means the application options. It is important to always end the administration session in this way. Otherwise, the i3D sessions will remain open and only a master will be able to close them.

6.1.2.1 Internal structure of an i3D database

I3D technology is based on the use of digital signatures and other cryptographic techniques in order to assure database integrity and non-repudiation. In this section, certain aspects of the internal structure and functioning of an i3D database are described in an introductory way, in order to justify the security requirements included in this document.

From this point, the term logical table will be used to refer to the set of records on which a database user performs reading, insertion, updating and deletion operations. Analogically, logical records will refer to records of a logical table.

6.1.2.1.1 I3D tables

In an i3D database there is no one-to-one correspondence among logical tables and physical tables, those that really reside in the database management system. On the contrary, for each logical table there are three physical tables:

- Session table: Contains information on all i3D sessions (closed or not) performed over the logical table.
- Historic record table: Stores all updates of each record of the logical table.
- Browsing table: Contains duplicated information of the last version of each record of the logical table, to perform SQL queries.

6.1.2.1.2 Starting an i3D session

Each time a user starts an i3D session with a logical table, two records are added to the session table: the session start entry and the session end entry. These records contain several control fields among which the session identifier (`sessionid` field) is

included. This identifier is different for all the i3D sessions started by the various users over the logical table.

When starting the i3D session, in addition, a 3DES random symmetric cryptographic key is generated. It is called the session key. This key is stored in the session start record asymmetrically enciphered with destination to the database masters, so that only the masters can know it.

The session start and end records are asymmetrically signed with the user's digital signature certificate (`signature` field). Besides, all records of the session table are linked in a way that a possible fraudulent intermediate session insertion or deletion would be detected when verifying the database integrity. This linkage is done in the following way:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.
- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

6.1.2.1.3 Operations over the logical table

Once an i3D session has been started, it is said that the session is active. This means that the user may perform SQL operations over the logical table records, and the performed operations will be associated to that session. Below it is described how these operations over the logical table affect the historic record table and the browsing table.

Insertion of a logical record

Causes the insertion of a record in the historic record table (historic record) that contains the data to store encoded in DER (`info` field) and other control fields. The new record includes information on the identifier of the active session (`sessionid` field). The entire record is symmetrically signed using the session key (`hmac` field).

Moreover, a record is added to the browsing table that is associated to the historic record (through the `hmac` field). This record contains part of the logical record data that is stored in non-encoded fields.

Logical record update

Causes the insertion of a historic record that contains the new data of the logical record and that is related to the previous historic record of the same logical record. The new record includes information on the identifier of the active session (`sidcurrent` field). The entire record is symmetrically signed using the session key (`hmac` field).

The browsing table record associated to the previous historic record is updated with the new data and is associated to the new historic record (through the `hmac` field).

Selection and retrieval of a logical record

SQL selection queries that the user demands are performed over the browsing table columns. Each record of this table corresponds to a logical record.

Once the desired record has been selected from the browsing table, the historic record table is accessed (through the `hmac` field value) in order to recover the last historic record associated to the logical record. The current value of the logical record is obtained from the historic record decoding the data stored in the info column.

This operation does not cause the insertion or modification of data in any table.

Deletion of a logical record

Causes the insertion of a historic record marked in a special way to indicate that the logical record has been deleted and therefore no more historic records are associated to it (`deleted` field). The new historic record includes information on the identifier of the active session (`sidcurrent` field). The entire record is symmetrically signed using the key session (`hmac` field).

Additionally, the entry corresponding to the logical record that was deleted is deleted from the browsing table, so that there is only a trace of its existence in the historic record table.

6.1.2.1.4 Normal closing of an i3D session

After performing a certain number of operations over the logical table, the user must eventually close the active i3D session. This way of finishing the session receives the name of normal closing.

When closing an i3D session, the session end record that was inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session (`hashchain` field) is added to this record. If the session only consisted on query operations then this field will remain empty.

The value of the `entrytype` field is also modified as indicated below. Once updated, the session end record is signed asymmetrically again, with the user's digital signature certificate and the session is considered closed.

The `entrytype` column of the session table allows distinguishing the session start record from the session end record and, in the second case, indicates the closing mode of the session.

6.1.2.2 Functional Requirements satisfied by Security Functions

The Secure Database services are composed of the following security functions:

- Database Integrity Verification Function (FUNC_DBIV). This functionality is able to detect modifications to the KeyOne database records (records containing certificates, CRLs, requests, audit logs, KeyOne batches, …). This verification is based on the KeyOne i3D mechanism that assures the integrity service and integrity verification service.

  This function basically consists of the i3dverify.ws command line tool that performs the verification from some certificates and keys according to the type of test to perform. The possible tests related to this function are the following:

  - Session integrity verification

This test consists on verifying historic records and header information associated to a certain i3D session. It must be used when it is suspected (or there is certainty) that a session presents inconsistencies.

- Integrity test of a closed session

  Through this test the integrity of all the historic records generated in a certain i3D session is verified, also checking that no records have been inserted or deleted fraudulently. The session must be closed (the session end record entrytype field must have a value greater than 1).

  Under this assumption the integrity of the session can be verified without knowing the session key, therefore any entity can run the test. The digital signature certificate of the user that performed the session is required. If the session was closed by a master, the master's digital signature certificate is also required.

- Integrity test of an open session

  In case of non-closed i3D sessions, it is also possible to perform the session integrity test. In order to do so, however, knowledge of the corresponding session key is required. Therefore, only a master can perform this test. Moreover, the digital signature certificate of the user that started the session is required.

  The integrity tests are advised to run when all database users are disconnected. This will assure that any session that remains open is an inactive session. However, this test can also be run over an active session.

  In order to run this test the verification tool first checks the integrity of the session start and end records, verifying its asymmetric signature (signature field).

- Integrity record table integrity verification

  This test allows verifying the full contents of a historic record table, through a sequential verification of all sessions contained in it. It is also possible to indicate that each session should be exhaustively verified as explained above.

  Through this test, the integrity of all i3D sessions performed over a certain logical table is verified, also checking that no intermediate sessions have been fraudulently inserted or deleted. It is necessary to know the digital signature certificates of all the users that have performed sessions over the table. Besides, if there are sessions that were closed by masters, the digital signature certificates of these masters must be known, as well.

- Check Database Capacity (FUNC_CDBC). This service allows manage the KeyOne services when the database capacity is full.

- Session Table Management (FUNC_I3DSESSION). This functionality is in charge of linking all the records of the session table by means the asymmetrical signature of the session start and end records.

- Historic Table Management (FUNC_I3DHISTORIC). This functionality is in charge of providing an integrity mechanism of the historic and browsing tables. This

mechanism consists of the generation of the symmetrical signature of the historic record, and the inclusion of this signature in the related record of the browsing table.

These services satisfy the following requirements:

6.1.2.2.1 FAU_STG.1.1 (iteration 2)

The TSF protects the stored audit records from unauthorized deletion because the KTS does not have any functionality to delete records from the audit database. From the KeyOne applications, it is not possibe to delete any registry from any database managed by these applications.

6.1.2.2.2 FAU_STG.1.2 (iteration 2)

As in the Database Integrity Verification Function section, page 107, has been explained, the FUNC_DBIV function is able to detect modifications to the database records, and therefore it allows detect modifications to the audit records.

6.1.2.2.3 FDP_SDI_CIMC.3.1

This requirement forces to provide of the integrity service (by means digital signatures, keyed hashes or authentication codes) to the public keys stored within the CIMC but not within a FIPS 140-1 validated cryptographic module.

The public key that has been certified is protected by means the digital signature related to the certificate. If the certificate is a root certificate, because the trusted certificates are stored in a i3D database, then the i3D integrity mechanisms provide the integrity security service.

In the communication between the RA and the CA of a public key not certified, the integrity of it is provided by means the signed format that is used for this communication (KeyOne batch) (FDP_SDI_CIMC.3.1 section, page 142), and for the integrity mechanism provided by the SSL/TLS protocol (FDP_SDI_CIMC.3.1 section, page 142). If the public key is stored in the database of the KeyOne system, then it is protected by means the integrity provided by the i3D database.

The Session Table Management Function (FUNC_I3DSESSION) generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, a possible fraudulent intermediate session modification can be detected when verifying the database integrity, by means the linkage of the records of the session table (the asymmetric signature of the session start record includes the `signature` field value of the previous session start record, and the asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record).

The Historic Table Management Function (FUNC_I3DHISTORIC) generates a symmetric digital signature (HMAC) of each record in the historic record table, using the session key. As it is explained in the Operations over the logical table section, page 106, additionally, when a record is added to the browsing table, it is associated to the historic record using the `hmac` field inserted in the related historic record.

6.1.2.2.4 FPT_STM.1.1 (iteration 2)

This requirement forces to provide reliable time stamps for its own use.

The reliability is provided by means using a time provided by the system clock that is synchronised by an NTP client installed in the same host where the KeyOne servers. This NTP component is synchronized with a reliable clock that obtains the Co-ordinated Universal Time from a reliable source. This communication is carried out by means the NTP (Network Time Protocol) protocol.

The stamping characteristic is accomplished because the i3D database (where the relationship between the data and the time is stored) provides the integrity service. The Session Table Management Function (FUNC_I3DSESSION) and the Historic Table Management Function (FUNC_I3DHISTORIC) provides the integrity functions that are possible the integrity functionality of the i3D database.

6.1.2.2.5 FPT_CIMC_TSP.1.1

This requirement forces to provide the creation of the following audit log signing event:

- It must compute a digital signature, keyed hash, or authentication code over the entries in the audit log.
- This digital signature, keyed hash, or authentication code must be computed over, at least, every entry that has been added to the audit log since the previous audit log signing event and the digital signature, keyed hash, or authentication code from the previous audit log signed event.
- The digital signature, keyed hash, or authentication code from the audit log-signing event shall be included in the audit log.

This requirement is compliant by means the Session Table Management Function (FUNC_I3DSESSION). This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, there is the following linkage of the records of the session table:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.
- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

When an i3D session is closed, then the session end record that is inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session is added to this record. Once updated, the session end record is signed asymmetrically again with the user's digital signature certificate and the session is considered session.

6.1.2.2.6 FPT_CIMC_TSP.1.2

This requirement forces to provide the creation of the following audit log signing event:

- It must compute a digital signature, keyed hash, or authentication code over the entries in the audit log.
- This digital signature, keyed hash, or authentication code must be computed over, at least, every entry that has been added to the audit log since the previous audit log signing event and the digital signature, keyed hash, or authentication code from the previous audit log signed event.
- The digital signature, keyed hash, or authentication code from the audit log-signing event shall be included in the audit log.

This requirement is compliant by means the Session Table Management Function (FUNC_I3DSESSION). This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, there is the following linkage of the records of the session table:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.
- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

When an i3D session is closed, then the session end record that is inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session is added to this record. Once updated, the session end record is signed asymmetrically again with the user's digital signature certificate and the session is considered session.

6.1.2.2.7 FPT_CIMC_TSP.1.4

This requirement forces to provide the creation of the following audit log signing event:

- It must compute a digital signature, keyed hash, or authentication code over the entries in the audit log.
- This digital signature, keyed hash, or authentication code must be computed over, at least, every entry that has been added to the audit log since the previous audit log signing event and the digital signature, keyed hash, or authentication code from the previous audit log signed event.
- The digital signature, keyed hash, or authentication code from the audit log-signing event shall be included in the audit log.

This requirement is compliant by means the Session Table Management Function (FUNC_I3DSESSION). This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, there is the following linkage of the records of the session table:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.
- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

When an i3D session is closed, then the session end record that is inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session is added to this record. Once updated, the session end record is signed asymmetrically again with the user's digital signature certificate and the session is considered session.

6.1.2.2.8 FAU_STG.4.1

This requirement forces to prevent auditable events, except those taken by the Auditor, if the audit trail is full.

This requirement specifies the behaviour of the TOE if the audit trail is full. The requirement also states that no matter how the requirement is instantiated, the authorised user with specific rights to this effect (Auditor), can continue to generate auditable events (actions). The reason is that otherwise the authorised user could not even reset the system.

In this case, if the audit trail is full, the Check Database Capacity Function (FUN_CDBC) manages the control of the situation. This function is in charge of the following tasks:

- Generation of a blinded auxiliary log in disc.
- Roll-back of the performed actions related to the event.
- The system is stopped.

When an administrator tries to starts the system, then the database capacity is checked (only will be successful started until the database has assigned with the necessary capacity to generate the initial log entries). Because the KeyOne servers are shutdown, then no more log registries are generated related to auditable events.

The requirement state that the authorised user with specific rights (Auditor), can continue to generate auditable events, in order to reset the system or repair the database capacity problem. In this case, the Auditor role can perform the following tasks:

- Examine the auxiliary log.
- Examine the audit table from the Database Management System.

Otherwise, starting the KeyOne server, the Auditor role will cannot have more information about the situation in order to repair the problem.

6.1.2.2.9 FPT_CIMC_TSP.1.3

This requirement forces to allow configure the frequency at which the audit log-signing event occurs.

The audit log-signing event is generated by the Session Table Management Function (FUNC_I3DSESSION). This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, there is the following linkage of the records of the session table:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.

- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

When an i3D session is closed, then the session end record that is inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session is added to this record. Once updated, the session end record is signed asymmetrically again with the user's digital signature certificate and the session is considered session.

Additionally, when an historic record is added (insertion, update or deletion of a logical record), a symmetrical signature of this historic record is generated and it is added into the historic table and also into the related browsing record. The generation of the symmetrical signature of the database records is performed by the FUNC_I3DHISTORIC function.

Because for each modification (addition, update or delete) of a database registry, the i3D mechanism assures the generation of a digital signature that guarantees the database integrity, then KeyOne system works as if it was configured at the maximum frequency, and therefore the frequency most secure (refinement of the FPT_CIMC_TSP.1.3 requirement).

6.1.2.2.10 FDP_CIMC_BKP.2.1

This requirement forces to protect the backup data against modification through the use of digital signatures, keyed hashes, or autentication codes.

The KeyOne System includes a functionality of backup that is in charge of backing up the whole KeyOne system necessary to reconstruct the current status from this backup and a copy of the same version of the software used to install initially the KeyOne system. The data stored in the system backup necessary to recreate the state of the system at the time the backup includes all the information stored in the KeyOne Databases. This information is protected by means the KeyOne i3D mechanism and making use of the FUNC_I3DSESSION and FUNC_I3DHISTORIC security functions.

The audit log-signing event is generated by the Session Table Management Function (FUNC_I3DSESSION). This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session section, page 105, there is the following linkage of the records of the session table:

- The asymmetric signature of the session start record includes the `signature` field value of the previous session start record.

- The asymmetric signature of the session end record includes the value of the `signature` field of the corresponding session start record.

When an i3D session is closed, then the session end record that is inserted in the session table when the session was started is modified. Specifically, the accumulated hash of all the historic records generated during the session is added to this record. Once updated, the session end record is signed asymmetrically again with the user's digital signature certificate and the session is considered session.

Additionally, when an historic record is added (insertion, update or deletion of a logical record), a symmetrical signature of this historic record is generated and it is added into the historic table and also into the related browsing record. The

generation of the symmetrical signature of the database records is performed by the FUNC_I3DHISTORIC function.

Because for each modification (addition, update or delete) of a database registry, the i3D mechanism assures the generation of a digital signature that guarantees the database integrity, then KeyOne part of the backup that includes the information stored in the KeyOne Database, is protected agains modification through the use of digital signatures.

6.1.3 Access Control Management

KeyOne technology uses an access control based on roles management when a user tries to access to TOE functions managing KeyOne resources.

Depending on the security policy, the names of the KeyOne roles can be different. Thus, if the CIMC security policy is used, the Administrator role would be the System Administrator and Security Officer for the CWA policy; the Officers role for the CIMC would be the Registration Officer role for the CWA policy; while the Auditors role for the CIMC policy would be the System Auditor role for the CWA policy.

System Operator is not a KeyOne role in the strict sense, since there is no application function assigned to this role. In fact, when talking of users with the System Operator role we do not refer to users registered as such in the KeyOne system; it is a way to designate users who can supply secrets required to run the application. From this point of view you can consider System Operators as mere resources whose presence is required to start the application.

6.1.3.1 Users, groups and roles

KeyOne application users belong to one or more groups and they are both defined in the whole KeyOne system (i.e. in all applications forming the system). To each group of users one or more roles, which are specific for each application, can be assigned. It is not possible to directly assign roles to individual users. Each role, in a KeyOne application, represents a set of specific permissions over functions belonging to this application. This set of permissions is established (i.e. is granted) by loading the security policy, which is selected during the KeyOne Console start-up.

All application users and groups of users that make up the KeyOne system must be created from the KeyOne Console application. However, KeyOne Console can only assign roles to those groups, which are defined in KeyOne Console. Therefore, once system users and groups have been created, role assigning from different applications must be performed from each one. This is, role assigning from a KeyOne CA instance must be performed from that particular KeyOne CA instance.

All KeyOne system application **users** must be created from the KeyOne Console application. This way, KeyOne Console is used to register all system users and not only those which are specific users for KeyOne Console.

A **group of users** is a set of users to which roles from different KeyOne system applications can be assigned. This way, user groups constitute the mechanism to assign roles to system users. This way, a user holds, in each one of the KeyOne system applications, all the roles that the group to which he/she belongs to holds.

KeyOne system user groups must be created from the KeyOne Console application. This way, KeyOne Console is used to register al system groups and not only those KeyOne Console specific groups.

There are certain user groups that are treated differently:

- Initial groups

  Initial groups are groups that are created by loading the security policy during the KeyOne Console start-up. Later, it is possible to add additional groups, except if the policy does not allow it.

- Main groups belonging to KeyOne applications

  KeyOne applications main groups are subsets of initial groups for which the following restrictions are established:

  - The main KeyOne Console groups must always have at least one enabled user.
  - Both for KeyOne Console main groups as for the rest of the KeyOne applications, it is not possible to the roles that the security policy assigns. The policy can allow assigning additional roles to the main groups. However, the system will not allow to delete any of the roles that the policy has assigned them.

  The security policy selected during the KeyOne Console start-up defines the main groups of the system and of each one of the policies. Every policy defines, at least, the following main groups:

  - For KeyOne Console:
    - *Administrators* group, usually called **System Administrators** (`ADMIN_GROUP`).
    - *Security officers* group, usually called **Security Officers** (`MAIN_GROUP`).

    Both users that intervene in the KeyOne Console start-up (i.e. initial users) is automatically assigned to the rest of the groups.
  - For the rest of the KeyOne applications:
    - *Administrators* group, usually called **System Administrators** (`ADMIN_GROUP`).
    - *Security officers* group, usually called **Security Officers** (`MAIN_GROUP`).

    The user that creates a KeyOne application different from KeyOne Console must belong to the second group. A KeyOne application main group can be the same as those in KeyOne Console (i.e. have the same name).

All KeyOne Console users hold one or more **roles**. Theses roles are part of the KeyOne Console configuration and are initialized from the values defined by the security policy selected during the start-up.

It is not possible to directly assign roles to individual users, but to groups. This way, users hold roles according to those assigned to the groups they belong to.

Each role in KeyOne Console represents a set of specific permissions over application functions. This set of permissions is established when loading the security policy selected during the KeyOne Console start-up, and cannot be modified once it has been established.

A KeyOne Console user can have more than one role whenever roles are not incompatible among them.

Each one of the roles has a specific aim defined in KeyOne Console and, consequently, holds a set of specific privileges to execute the KeyOne Console functions.

It is possible to define incompatibilities among roles in order to avoid that a user can access all KeyOne Console functions.

6.1.3.2 Controlling the access to the KeyOne functions

The control access performed by the KeyOne applications is based on the fact that the user could execute a certain operation.

The relationship between KeyOne soft-pages and operations is maintained in a signed configuration file, and the relationship between the operations and roles is established in the security policy.

The KeyOne system maintains ACLs (Access Control Lists) managed by the KeyOne applications:

- When the application is loaded, the ACL object contains information about this application (e.g. roles, users, relationships between operations and roles, …).
- In the login process, this object contains user information (e.g. the role/s assigned to the user).

The winscryptor engine performs the association between KeyOne soft-pages and operations, and it loads this information in memory.

The TestAction/CheckAction function (method belonging to the ACL object) uses the information of the ACL object and the winscryptor, and it determines if the user can or not execute a certain KeyOne soft-page. All the KeyOne functions that can be accessed by a user, must execute the TestAction/CheckAction method.

Regarding to the winscryptor actions, they always must execute the TestAction/CheckAction method (the winscryptor always execute the .runMethod method, it invoke the testAllowed function, and this function always invokes the TestAction/CheckAction function).

Regarding to the actions that can be personalized, they also must incorporate an invocation to the TestAction/CheckAction function. In order to successful execution of the personalized new programming code, the KeyOne server engine requires the invocation of the TestAction/CheckAction function.

6.1.3.3 Functional Requirements satisfied by Security Functions

The Access Control Management services are composed of the following security function:

- Access Control Function (FUNC_ACCESSCTRL). This functionality allows control the access to the TOE function by means the use of roles assigned to the user.

This service satisfies the following requirements:

6.1.3.3.1 FMT_MOF.1.1 (iteration 2)

The Access Control Function is able to restrict the functionality indicated in this requirement to the roles included in the table below:

Section/Function	Component	Function/Authorized Role
Security Audit		The capability to configure the audit parameters shall be restricted to Administrators.
Backup and Recovery		The capability to configure the backup parameters shall be restricted to Administrators. The capability to initiate the backup or recovery function shall be restricted to [assignment: *Administrator*]
Certificate Registration		The capability to approve fields or extensions to be included in a certificate shall be restricted to Officers. If an automated process is used to approve fields or extensions to be included in a certificate, the capability to configure that process shall be restricted to Officers.
Data Export and Output		The export of CIMC private keys shall require the authorization of at least two Administrators or one Administrator and one Officer, Auditor, or Operator.
Certificate Status Change Approval		Only Officers shall configure the automated process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall configure the automated process used to approve the placing of a certificate on hold or information about the hold status of a certificate.
CIMC Configuration		The capability to configure any TSF functionality shall be restricted to Administrators. (This requirement applies to all configuration parameters unless the ability to configure that aspect of the TSF functionality

		has been assigned to a different role elsewhere in this document).
Certificate Profile Management	FMT_MOF_CIMC.2 Certificate profile management FMT_MOF_CIMC.3 Extended certificate profile management	The capability to modify the certificate profile shall be restricted to Administrators.
Revocation Profile Management		The capability to modify the revocation profile shall be restricted to Administrators.
Certificate Revocation List Profile Management	FMT_MOF_CIMC.4 Certificate revocation list profile management FMT_MOF_CIMC.5 Extended certificate revocation list profile management	The capability to modify the certificate revocation list profile shall be restricted to Administrators.
Online Certificate Status Protocol (OCSP) Profile Management	FMT_MOF_CIMC.6 OCSP profile management	The capability to modify the OCSP profile shall be restricted to Administrators.

Table 6-1. Authorized Roles for Management of Security Functions Behaviour

The relationships between the operations and roles are established in the security policy file. The table above affects to the following operations/privileges (the role that can access to these operations can be configured in the security policy file):

- Configure the audit parameters: `VIEW_DATABASETREE`, `CREATE_DATABASE_TABLES`, `RENAME_DATABASE_TABLES`, `CHANGE_DATABASE_CONNECTION`, `VIEW_REGISTEREDDATABASES`, `EDIT_REGISTEDDATABASES`, `EDIT_LOGS_REGISTER`.
- Configure the backup parameters: `EXECUTE_SYSTEM_BACKUP`.
- Initiate the backup or recovery function: `EXECUTE_SYSTEM_BACKUP`.
- Approve fields or extensions to be included in a certificate: `MODIFY_PROFILES`.
- Exportation of CIMC private keys: `ISSUE_CERTIFICATES`, `KEY_RECOVERY`. The exportation of the Keys to the Hardware Security Module is performed when the KeyOne system is started. In this case the operator that introduces the correct authentication cards is the one that can start the system and therefore that can export CIMC private keys to the HSM.
- Configure the automated process used to approve the revocation of a certificate or information about the revocation of a certificate. This privilege does not apply because in KeyOne technology does not exist an automated process designated to this purpose.

- Configure the automated process used to approve the placing of a certificate on hold or information about the hold status of a certificate. This privilege does not apply because in KeyOne technology does not exist an automated process designated to this purpose.

- Configure any TSF functionality: `BROWSE_USER`, `CREATE_USERS`, `DELETE_USERS`, `ENABLE_USER`, `VIEW_USER_PROPERTIES`, `EDIT_USER_PROPERTIES`, `BROWSE_GROUPS`, `CREATE_GROUP`, `DELETE_GROUP`, `VIEW_GROUP_PROPERTIES`, `EDIT_GROUP_PROPERTIES`, `EDIT_PASSWORD_RULES`, `VIEW_PASSWORD_RULES`, `EDIT_SYSTEM_CERTIFICATES`, `VIEW_SYSTEM_CERTIFICATES`, `EDIT_LOGON_TIMEOUT`, `VIEW_LOGON_TIMEOUT`, `VIEW_APP_CERTS`, `VIEW_APP_KEYS`, `VIEW_APP_CRLS`, `INSTALL_APP_ROOT`, `INSTALL_APP_CERT`, `REMOVE_APP_CERT`, `EXPORT_APP_CERT`, `INSTALL_APP_CRL`, `REMOVE_APP_CRL`, `EXPORT_APP_CRL`, `GENERATE_KEY_PARAMS`, `VIEW_APP_KEY_DEFS`, `EDIT_APP_KEY_DEFS`, `RENEW_APP_KEYS`, `GENERATE_APP_KEYS`, `REMOVE_APP_KEYS`, `CREATE_APPLICATION`, `SELECT_APP_CERTS`, `VIEW_CERT`, `MODIFY_PROFILES`, `VIEW_PERMISSIONS`, `VIEW_ROLE_COMPATIBILITIES`, `EDIT_ROLE_COMPATIBITIES`, `VIEW_USER_ROLES`, `EDIT_USER_ROLES`.

- Modify the certificate profile: `MODIFY_PROFILES`.

- Modify the revocation profile: `MODIFY_PROFILES`.

- Modify the certificate revocation list profile: `MODIFY_PROFILES`.

- Modify the OCSP profile: `MODIFY_OCSP_PROFILES`.

By modifying the KeyOne security policy is possible to set these privileges to the appropriate role that will be able to execute the related operation.

The Access Control Function by means the TestAction/CheckAction method is able to control the access to the operations by using the roles assigned to the user can execute the action.

6.1.3.3.2 FDP_ACC.1.1 (iteration 2)

To enforce the security policy, the access control of the KeyOne system is based on the following secure relationships:

- KeyOne soft-pages are related to operations in a signed configuration file (`pssmanager.actions`).

- Operations are related to roles in the KeyOne security policy (`policies`).

The TestAction/CheckAction function determines the access of a user to a function by using the information loaded in the ACL object (roles assigned to the current user and relationships between operations and roles). TOE enforces access control policy on the following entities and objects:

- Users of the KeyOne applications.

- Resources managed by the system.

- Privileges defined by the system and that can be assigned to the application roles.

6.1.3.3.3 FDP_ACF.1.1 (iteration 2)

To enforce the security policy, the access control of the KeyOne system is based on the following security attributes:

- Identity of the subject.
- Set of roles that the subject is authorized to assume.

The Access Control Function is based on the ACL object in order to determine the access of a user to the execution of a KeyOne function. In the login process, the ACL object is loaded with the necessary user information (user identification and role/s assigned to the user).

6.1.3.3.4 FDP_ACF.1.2 (iteration 2)

The KeyOne Access Control can be configured in order to conform the following rules specified in the table below:

Section/Function	Component	Function/Authorized Role
Certificate Request Remote and Local Data Entry		The entry of certificate request data shall be restricted to Officers and the subject of the requested certificate.
Certificate Revocation Request Remote and Local Data Entry		The entry of certificate revocation request data shall be restricted to Officers and the subject of the certificate to be revoked.
Data Export and Output		The export or output of confidential and security-relevant data shall only be at the request of authorized users
Key Generation	FCS_CKM.1 Cryptographic Key Generation	The capability to request the generation of Component keys (used to protect data in more than a single session or message) shall be restricted to Administrators.
Private Key Load		The capability to request the loading of Component private keys into cryptographic modules shall be restricted to Administrators.
Private Key Storage		The capability to request the decryption of certificate subject private keys shall be restricted to Officers. The TSF shall no provide a capability to decrypt certificate subject private keys that may be used to generate digital signatures. At least two Officers or one Officer and an Administrator, Auditor, or Operator shall be required to request the decryption of a certificate subject private key.

Trusted Public Key Entry, Deletion, and Storage		The capability to change (add, revise, delete) the trusted public keys shall be restricted to Administrators.
Secret Key Storage		The capability to request the loading of CIMC secret keys into cryptographic modules shall be restricted to Administrators.
Private and Secret Key Destruction		The capability to zeroize CIMC plaintext private and secret keys shall be restricted to Administrators, Auditors, Officers, and Operators.
Private and Secret Key Export		The capability to export a component private key shall be restricted to Administrators. The capability to export certificate subject private keys shall be restricted to Officers. The export of a certificate subject private key shall require the authorization of at least two Officers or one Officer and an Administrator, Auditor, or Operator.
Certificate Status Change Approval		Only Officers and the subject of the certificate shall be capable of requesting that a certificate be placed on hold. Only Officers shall be capable of removing a certificate from on hold status. Only Officers shall be capable of approving the placing of a certificate on hold. Only Officers and the subject of the certificate shall be capable of requesting the revocation of a certificate. Only Officers shall be capable of approving the revocation of a certificate and all information about the revocation of a certificate.

Table 6-2. Access Controls

The table above affects to the following operations/privileges (the role that can access to these operations can be configured in the security policy file):

- Entry of certificate request data: `ISSUE_CERTIFICATES`.
- Entry of certificate revocation request data: `BROWSE_CERTS_DB`, `REVOKE_CERTIFICATES`.
- Export or output of confidential and security-relevant data: `ISSUE_CERTIFICATES`, `KEY_RECOVERY`. The exportation of the Keys to the

Hardware Security Module is performed when the KeyOne system is started. In this case the operator that introduces the correct authentication cards is the one that can start the system and therefore that can export CIMC private keys to the HSM.

- Request the generation of Component keys (used to protect data in more than a single session or message): `GENERATE_KEY_PARAMS`, `RENEW_APP_KEYS`, `GENERATE_APP_KEYS`.
- Request the loading of Component private keys into cryptographic modules. The exportation of the Keys to the Hardware Security Module is performed when the KeyOne system is started. In this case the operator that introduces the correct authentication cards is the one that can start the system and therefore that can load the component private keys into cryptographic modules.
- Request the decryption of certificate subject private keys: `KEY_RECOVERY`.
- The TSF shall not provide a capability to decrypt certificate subject private key that may be used to generate digital signatures. The KeyOne Key Archive component (located in the KeyOne CA product) allows filtering the kind of certificate (not digital signature certificates) that can be archive in the Key Archive databases (in order to recovery later the subject private key).
- Request the decryption of a certificate subject private key: `KEY_RECOVERY` The KeyOne Key Archive application requires the presence of two roles in order to recover the subject private key.
- Change the trusted public keys: `EDIT_SYSTEM_CERTIFICATES`, `VIEW_SYSTEM_CERTIFICATES`.
- Request the loading of CIMC secret keys into cryptographic modules: The exportation of the Keys to the Hardware Security Module is performed when the KeyOne system is started. In this case the operator that introduces the correct authentication cards is the one that can start the system and therefore that can load the component private keys into cryptographic modules.
- Zeroize CIMC plaintext private and secret keys: The TOE does not never has the CIMC private and secret keys in plaintext, and therefore the zeroize function applied to the CIMC plaintext private and secret keys is not applicable. The KeyOne system provides access to the Hardware Security Module, and consequently it manages the CIMC private and secret keys. The TOE relies on the FIPS 140-2 validated module to perform critical key generation, key storage and zeroization for key destruction. No CIMC private and secret keys are stored in the KeyOne system, and the KeyOne system accesses the HSM to perform operations with this kind of keys.
- Export a component private key: `ISSUE_CERTIFICATES`, `KEY_RECOVERY`. The exportation of the Keys to the Hardware Security Module is performed when the KeyOne system is started. In this case the operator that introduces the correct authentication cards is the one that can start the system and therefore that can export CIMC private keys to the HSM.
- Export certificate subject private keys: `ISSUE_CERTIFICATES`, `KEY_RECOVERY`. The exportation of certificate subject private keys requires the authorisation of two roles.

- Request a certificate to be place on hold: BROWSE_CERTS_DB, REVOKE_CERTIFICATES.
- Remove a certificate from a hold status: BROWSE_CERTS_DB, REVOKE_CERTIFICATES.
- Approve the placing of a certificate on hold: BROWSE_CERTS_DB, REVOKE_CERTIFICATES.
- Request the revocation of a certificate: BROWSE_CERTS_DB, REVOKE_CERTIFICATES.
- Approve the revocation of a certificate and all information about the revocation of a certificate: BROWSE_CERTS_DB, REVOKE_CERTIFICATES.

The Access Control Function is based on the ACL object in order to determine the access of a user to the execution of a KeyOne function. In the login process, the ACL object is loaded with the necessary user information (user identification and role/s assigned to the user).

6.1.3.3.5 FMT_MOF_CIMC.3.2

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for the certificate fields and extensions. The MODIFY_PROFILES privilege can be assigned to an specific role in the KeyOne security policy.

When the certification template configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the MODIFY_PROFILES privilege.

6.1.3.3.6 FMT_MOF_CIMC.3.3

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for the certificate fields and extensions. The MODIFY_PROFILES privilege can be assigned to an specific role in the KeyOne security policy.

When the certification template configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the MODIFY_PROFILES privilege.

6.1.3.3.7 FMT_MOF_CIMC.3.4

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for the certificate fields and extensions. The MODIFY_PROFILES privilege can be assigned to an specific role in the KeyOne security policy.

When the certification template configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the MODIFY_PROFILES privilege.

6.1.3.3.8 FMT_MOF_CIMC.5.2

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for the CRLs fields and extensions. The MODIFY_PROFILES privilege can be assigned to an specific role in the KeyOne security policy.

When the CRL template configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the MODIFY_PROFILES privilege.

6.1.3.3.9 FMT_MOF_CIMC.5.3

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for the CRLs fields and extensions. The MODIFY_PROFILES privilege can be assigned to an specific role in the KeyOne security policy.

When the CRL template configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the MODIFY_PROFILES privilege.

6.1.3.3.10 FMT_MOF_CIMC.6.2

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for some fields of the OCSP response messages.

The EDIT_VA_CONFIG privilege can be assigned to an specific role in the KeyOne security policy. When the OCSP message configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the EDIT_VA_CONFIG privilege.

6.1.3.3.11 FMT_MOF_CIMC.6.3

The KeyOne system can be configured in order to determine that an specific role could assign the acceptable values for some fields of the OCSP response messages.

The EDIT_VA_CONFIG privilege can be assigned to an specific role in the KeyOne security policy. When the OCSP message configuration function is invoked, then the Access Control Function will check the roles related to the user, and the roles assigned to the EDIT_VA_CONFIG privilege.

6.1.3.3.12 FPT_RVM.1.1 (iteration 2)

The TestAction/CheckAction function is responsible of the management of the KeyOne access control service. This method (belonging to the ACL object) uses the information of the ACL object and the winscryptor, and it determines if the user can or not execute a certain KeyOne soft-page. All the KeyOne functions that can be accessed by a user, must execute the TestAction/CheckAction method.

Regarding to the winscryptor actions, they always must execute the TestAction/CheckAction method (the winscryptor always execute the .runMethod method, it invoke the testAllowed function, and this function always invokes the TestAction/CheckAction function).

Regarding to the actions that can be personalized, they also must incorporate an invocation to the TestAction/CheckAction function. In order to successful execution of the personalized new programming code, the KeyOne server engine requires the invocation of the TestAction/CheckAction function.

6.1.3.3.13 FDP_ACF.1.3 (iteration 2)

This requirement does not imply the use of any access control mechanism, and therefore there are not any TOE security function related to it.

6.1.3.3.14 FDP_ACF.1.4 (iteration 2)

This requirement does not imply the use of any access control mechanism, and therefore there are not any TOE security function related to it.

6.1.4 Identification and Authentication

Identification and authentication processes are required before starting any KeyOne application. The TOE keeps information (users, passwords, certificates, …) related to these processes in a secure repository (Private Secure Store) offering the integrity and confidential (for sensitive data) services.

The authentication processes carried out by the KeyOne server is provided by means security mechanisms explained in the Secure Communications section, page 135.

6.1.4.1 Authentication of the initial users

The authentication of the initial Administrator and Security Officer is carried out during the system start-up.

Authentication of the initial KeyOne system Administrator

Initially the Administrator uses the installation assistant in order to introduce the basic configuration (cryptographic module of the system, database of the system and card reader of the user). At the end of the process, the assistant requests a password, and the basic configuration is stored ciphered by means this password. After, in the initialisation phase, the Security Officer asks to the Administrator for this password in order to load the basic configuration. When the initialisation phase is finished, the Administrator will be established as system user (authentication using password).

Authentication of the initial KeyOne system Security officer

This authentication can be carried out by means the following two options:

a) The Security Officer owns either an smart card initialised by Safelayer (it is delivered with the CD) or issued by another CA.

b) The Security Officer does not have an smart card. The first time that he enters to the system, it is necessary that he introduces a username and a password in order to authenticate him subsequently. This option is not possible if the security policy forbids the use of passwords.

6.1.4.2 Special groups of users

There are the following groups of users that are managed in a special way:

6.1.4.2.1 Groups defined by the policy

The security policy defines a minimum set of user groups that will have the system and each one of the applications. It is possible to define more groups (if it is allowed by the policy), but it is no possible to eliminate or rename the groups defined by the policy.

6.1.4.2.2 Main groups

The security policy defines the main groups for the system and for each application. These groups are a subset of the ones defined by the policy, and they are managed in a special way. As a minimum, the policy defines the following main groups:

- For the system: Administrators and Security Officers. The two users that are involved in the system start-up are assigned automatically to these groups.
- For each application: Administrators and Security Officers (can be the same groups that the main groups for the system). The user that creates the application must belong to the second group.

The following restrictions are applied to these groups:

- The main groups of the system always must have an authorized user.

  This restriction guarantees that always it is possible to starts the KeyOne Console in order to resolve certain start problems. In case of the main groups of the applications, this restriction is not necessary because it is always possible to enter in KeyOne Console and to assign users.

- It is not possible to reduce the roles assigned by the policy to the main groups.

  This restriction guarantees that the main groups always will be authorized in order to carried out the required tasks in case of start problems. In this case, the restriction is also applied to the main groups of the applications, because certain start problems must be resolved by entering to the application in fault tolerant mode, and the assignment of roles to groups must be carried out in the application.

6.1.4.3 Authentication modes

The following user authentication modes are defined:

- Certificate (smart card)
- Username and password
- Username and security password. This mode can only be used by a Security Officer in order to start the KeyOne Console. This security password is automatically generated and it is exported to a file during the system start-up phase. This password must be stored in a secure way by means an external procedure. Nobody can know the password until it is recovered for using; in this moment it is re-generated and it must be again stored in a secure way.

The selected security policy can restrict the authentication modes allowed by certain users. For this reason, the security policy specifies one of the following configurations:

a) Authentication by means allowed password

b) Authentication by means forbidden password

6.1.4.3.1 Authentication by means allowed password

This configuration allows the authentication by means password for all users, using the following restrictions:

- It is necessary that always there is a user belonging to the system Administrators main group, and that this user has authorized the authentication by using a password (besides this type of authentication, an authentication by using certificate can be used).

  This guarantees that always an Administrator will can enter to the KeyOne Console in order to solve problems (e.g. reconfigure the smart card for authenticating users).

- It is necessary that always there is a user belonging to the system Security Officers main group, and that this user has authorized the authentication by using a password (besides this type of authentication, an authentication by using certificate can be used). Another possibility is to use a security password stored in a secure way.

  This guarantees that always a Security Officer will can enter to the KeyOne Console in order to solve problems (e.g. lost of his smart card, expiration of his certificate, …).

- The rest of users can use either the authentication by means certificate or the authentication by means password.

6.1.4.4 Functional Requirements satisfied by Security Functions

The Identification and Authentication services are composed of the following security functions:

- User Identification and Authentication Function (FUNC_UIDAUT). This functionality is able to identify and authenticate the user by means a username and a password/certificate previously assigned to the user.

These services satisfy the following requirements:

6.1.4.4.1 FIA_UAU.1.1 (iteration 2)

Depending on how the user has been configured he will be able to carry out the authentication through password or through proof of possession (cryptographic card)[6]. The authentication procedure that will be used must indicate by selecting the appropriate value in the Authentication mode list. The contents of the login screen will change depending on the value that has been selected.

[6] *Authentication through certificate can only be selected if a card-reader has been configured as the system's primary card-reader.*

Authentication through password

For indicating this option, the user must select the "`password`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Name of the user (`User name` mandatory field).

b) Password of the user (`Password` mandatory field).

The system will verify the username (identification) and password (authentication) that have been typed in.

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the name (identification failure) or the password (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to retype the name and password (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the authentication phase.

Authentication through certificate

For indicating this option, the user must select the "`card`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Introduction of the card in the card-reader that has been configured as the system's primary card-reader.

b) Card's PIN (`PIN` mandatory field).

The system will validate the user certificate (identification: the introduced certificate is a certificate that the system has been registered as a certificate of an authorised user), and it will verify the possession of the private key associated to it by means a Proof of Possession mechanism (authentication).

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the certificate (identification failure) or the Proof of Possession (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to introduce the certificate and related PIN again (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the authentication phase.

6.1.4.4.2 FIA_UAU.1.2 (iteration 2)

Depending on how the user has been configured he will be able to carry out the authentication through password or through proof of possession (cryptographic card)[7]. The authentication procedure that will be used must indicate by selecting the appropriate value in the Authentication mode list. The contents of the login screen will change depending on the value that has been selected.

Authentication through password

For indicating this option, the user must select the "`password`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Name of the user (`User name` mandatory field).

b) Password of the user (`Password` mandatory field).

The system will verify the username (identification) and password (authentication) that have been typed in.

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started

[7] *Authentication through certificate can only be selected if a card-reader has been configured as the system's primary card-reader.*

- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the name (identification failure) or the password (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to retype the name and password (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the authentication phase.

Authentication through certificate

For indicating this option, the user must select the "`card`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Introduction of the card in the card-reader that has been configured as the system's primary card-reader.

b) Card's PIN (PIN mandatory field).

The system will validate the user certificate (identification: the introduced certificate is a certificate that the system has been registered as a certificate of an authorised user), and it will verify the possession of the private key associated to it by means a Proof of Possession mechanism (authentication).

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the certificate (identification failure) or the Proof of Possession (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to introduce the certificate and related PIN again (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the authentication phase.

6.1.4.4.3 FIA_UID.1.1 (iteration 2)

Depending on how the user has been configured he will be able to carry out the identification through username or through certificate (cryptographic card)[8]. The identification procedure that will be used must indicate by selecting the appropriate value in the Identification mode list. The contents of the login screen will change depending on the value that has been selected.

Identification through username (authentication through password)

For indicating this option, the user must select the "`password`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Name of the user (`User name` mandatory field).

b) Password of the user (Password mandatory field).

The system will verify the username (identification) and password (authentication) that have been typed in.

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the name (identification failure) or the password (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to retype the name and password (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the identification phase.

[8] *Authentication through certificate can only be selected if a card-reader has been configured as the system's primary card-reader.*

Identification through certificate (authentication through proof of possession)

For indicating this option, the user must select the "`card`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Introduction of the card in the card-reader that has been configured as the system's primary card-reader.

b) Card's PIN (PIN mandatory field).

The system will validate the user certificate (identification: the introduced certificate is a certificate that the system has been registered as a certificate of an authorised user), and it will verify the possession of the private key associated to it by means a Proof of Possession mechanism (authentication).

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the certificate (identification failure) or the Proof of Possession (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to introduce the certificate and related PIN again (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the identification phase.

6.1.4.4.4 FIA_UID.1.2 (iteration 2)

Depending on how the user has been configured he will be able to carry out the identification through username or through certificate (cryptographic card)[9]. The identification procedure that will be used must indicate by selecting the appropriate value in the Identification mode list. The contents of the login screen will change depending on the value that has been selected.

[9] *Authentication through certificate can only be selected if a card-reader has been configured as the system's primary card-reader.*

Identification through username (authentication through password)

For indicating this option, the user must select the "`password`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Name of the user (`User name` mandatory field).

b) Password of the user (Password mandatory field).

The system will verify the username (identification) and password (authentication) that have been typed in.

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the name (identification failure) or the password (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to retype the name and password (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the identification phase.

Identification through certificate (authentication through proof of possession)

For indicating this option, the user must select the "`card`" value from the `Authentication mode` list.

In this step, the system will request for the following identification/authentication information:

a) Introduction of the card in the card-reader that has been configured as the system's primary card-reader.

b) Card's PIN (PIN mandatory field).

The system will validate the user certificate (identification: the introduced certificate is a certificate that the system has been registered as a certificate of an authorised user), and it will verify the possession of the private key associated to it by means a Proof of Possession mechanism (authentication).

If they are both correct:

- If the application is KeyOne Console, the main application screen is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started
- For the rest of the KeyOne applications:
    - The screen for selecting the application instance with which to start a session is shown.
    - The main screen of the application is shown and a session in which the user will be able to perform all functions that are allowed for each of the roles will be started.

However, if the certificate (identification failure) or the Proof of Possession (authentication failure) are erroneous, an error message will be shown on screen. In this case, the user will have to introduce the certificate and related PIN again (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE will prevent further authentication attempts).

In the login procedure, the user can abort the process before the completion of the identification phase.

6.1.4.4.5 FIA_USB.1.1 (iteration 2)

The User Identification and Authentication Function after identifying and authenticating the user, it associates the appropriate user security attributes (groups and roles) with subjects acting on behalf of that user.

When the KeyOne application starts, then the properties of this application are loaded in the ACL object. The security attributes related to the application, as the groups that are defined in the application, and the related roles linked with the defined groups, are loaded in the ACL object.

If the authentication mode is by using username and password, then when the user introduces the username, then the User Identification and Authentication Function indexes by using the SHA1 hash of the <username><password> the private secure store where the sensitive system information is stored. In this step, the function recovers the stored properties information of the user who is trying to login, and it loads in the ACL object the security information about the user, as the groups related to this user (in the ACL object is already stored the relationship between groups and roles, and therefore the relationship between the user and roles can be obtained).

If the authentication mode is by using a certificate, then when the user introduces the certificate and the PIN related to the smart card where it is stored, then the User Identification and Authentication Function indexes by using the SHA1 hash of the certificate the private secure store where the sensitive system information is stored. In order to authenticate the user, the system generates a random string (64 bytes) for requesting to the user a challenge-response proof. If the verification of the signature generated by the user with his private key is successful, then the function recovers the stored properties information of the user who is trying to login, and it loads in the ACL object the security information about the user, as the groups related to this user (in the ACL object is already stored the relationship between groups and roles, and therefore the relationship between the user and roles can be obtained).

6.1.5 Secure Communications

The implantation of secure mechanisms is required when a communication that affects to a component of the KeyOne TOE occurs. The TOE protects the data transfers either between KeyOne components, or between a KeyOne component and a TOE external component. This protection is achieved by means standards secure protocols (e.g. SSL/TLS protocol, OCSP, …), or by means the use of KeyOne proprietary protocols (e.g. KeyOne batches, NDCCP protocol, …).

6.1.5.1 KeyOne batches

Certificate generation and revocation services involve communication between KeyOne LRA and KeyOne CA. Messages exchanged during this communication process are called KeyOne batches and fulfil an specific syntax, which includes a digital signature in order to provide authentication, integrity and non-repudiation security services

Furthermore, these messages are transferred over an SSL/TLS connection. Therefore, confidentiality, authenticity and integrity of KeyOne LRA - KeyOne CA transactions are guaranteed.

Batches can be classified in two categories, depending on the type of request they come from:

- CR batches: Batches that contain a certification request.
- RR batches: Batches that contain a revocation, suspension or un-suspension request.

The batches are digitally signed by the issuer of the batch, and it will be verified in the receipt side by the recipient of the batch.

The KeyOne LRA application sends certification or revocation requests to the KeyOne CA application by using the KeyOne batch structure. The KeyOne CA application sends the generated certificates or the revocation results to the KeyOne LRA application by using the KeyOne batch structure.

These are the stages of a KeyOne batch life cycle:

- The batch is created by the RA. The RA sets the batch's generic information and adds the certification requests (or revocation requests) to the batch.
- For security reasons, the RA signs the batch after adding all the data to it. The RA's signature allows the CA to recognize the RA that sends the batch and makes sure that the batch has not been modified during the transmission.
- The RA sends the batch to the CA.
- The CA receives the batch, validates its signature and performs the operations requested.
- The CA adds the results of the operations to the batch and/or modifies the existing data. No new batch is generated. The data added/modified will depend on the operations requested. For a certification request, the generated certificates are added to the batch. For a revocation request, the revocation

result is added to the batch. The CA certification chain and CRLs are always added to the batch.

- For security reasons, the CA signs the batch after adding all the data to it. The CA signature allows the RA to recognize the CA that sends the batch and makes sure that the batch has not been modified during the transmission.
- The CA sends the batch to the RA.
- The RA receives the batch, validates its signatures and checks the results of the operations requested.
- If the batch contained certification requests, the RA extracts the certificates generated in response.

The KeyOne batch life cycle can be summarized as follows: the RA generates the batch and adds requests to it, the CA processes these requests and adds the responses to the batch. Therefore, the batch's structure can be seen as composed by the RA added-by data, the CA added-by data and the batch's generic information and batch extensions.

6.1.5.1.1 KeyOne LRA related data

- KeyOne Batch generic information

  The generic information identifies the batch, the type of its contents and its current status. The following fields make up the batch generic information:

  - `batchid`: Batch identifier. This field identifies a batch among all the batches generated by a Registration Authority.
  - `batchtype`: Batch type. This field identifies the type of batch requests. All the requests in a batch must have the same type. A batch can be:
    - `CR`: Certification batch: contains certification requests.
    - `RR`: Revocation batch: contains revocation requests.
  - `status`: Batch status. This field corresponds more or less with the stage of the life cycle the batch is in.

- KeyOne LRA related data

  The Registration Authority generates the batch and adds data to the batch to be processed by the CA. This data includes certification or revocation requests, as well as informational data to be exchanged.

  After this data has been added, the RA signs the batch, to guarantee that the CA receives it without any modification by a third party.

  The information that the RA adds to the batch is the following:

  - General Information:
    - `timereq`: Date and time when the batch was generated by the RA.
    - `rasubject`: Distinguished Name of the RA that generated the batch.

- `policiesReq`: List of all the certification policies requested. In a certification batch (CR), this list contains all the certification policy names that appear in certification requests. If the batch is not a CR batch, this field is empty.

- CSRs

  The most important part of the data added by the RA to the batch is the request list.

  - `csrReportSeq`: Request list. In a CR batch there will be certification requests. In an RR batch there will be revocation requests. In the other batch types this field can be empty.

- Arguments

  Arguments are additional data sent by the RA to the CA. This additional data can be information that the RA needs to retrieve once the CA has processed the batch, or other information that the CA may need. The fields are:

  - `scryptorGenericReq`: Data to be sent to the CA.

- Signature

  After adding all the data to the batch, the RA signs it to ensure that the CA receives the batch without modification of a third-party. The only field here is:

  - `rasignature`: Batch detached signature generated by the RA.

6.1.5.1.2 KeyOne CA related data

The batch is generated by the Registration Authority and sent to the CA. The CA reads the data in the batch, processes it and adds the results to the batch: certificates if the batch was a CR batch, or revocation results if the batch was an RR batch.

Like the RA, the CA also signs the batch after adding data, to ensure that the RA receives the batch without any modifications by a third party.

- Information

  Information fields identify the CA that processed the batch and when it was processed. The following fields make up the CA information:

  - `timeresp`: Date and time when the batch was processed by the CA.

  - `casubject`: Distinguished Name of the CA that processed the batch.

- Certificates

  The most important part of the data added by the CA to the batch is the response list. It is named "Certificates" because of the response to a CR batch (a certificates list), but it contains a revocation responses list if the processed batch is an RR batch.

- `certReportSeq`: Response list.

- Arguments

  The arguments are additional data sent by the RA to the CA. The CA will process this data and will send other data in response. The data that the CA sends can be the same data received (without modification) or any other data generated in response to the received data. The fields are:

  - `scryptorGenericGrant`: Data to be sent to the RA.

- Certification chain

  The CA adds its whole certification chain to the processed batch: its own certificates, the root certificates (if it is not a root itself), and all the certificates from the subordinate CAs between itself and the root CA (if any). The following fields make up the Certification chain:

  - `keyCertSignCertificates`: Certificate-signing certificates list. The combined certificates for certificate-signing and CRL-signing are also included here. If the CA is not a root CA, this list contains its own certificate and all the certificates from the subordinate CAs between itself and the root CA (if any). Otherwise, if the CA is a root CA, this list is empty.

  - `keyCertSignRootCertificate`: Certificate-signing certificate of the root CA. If the CA is a root CA, this certificate is its own certificate.

  - `crlSignOnlyCertificates`: CRL-signing certificates list. If the CA is not a root CA, this list contains its own certificate and all the certificates from the subordinate CAs between itself and the root CA (if any). Otherwise, if the CA is a root CA, this list is empty. This list is also empty if all the CAs have combined certificates for certificate-signing and CRL-signing.

  - `crlSignOnlyRootCertificate`: CRL-signing certificate of the root CA. If the CA is a root CA, this certificate is its own certificate.

  - `digitalSignatureCertificates`: Digital signature certificates list. If the CA is not a root CA, this list contains its own certificate and all the certificates from the subordinate CAs between itself and the root CA (if any). Otherwise, if the CA is a root CA, this list is empty.

  - `digitalSignatureRootCertificate`: Digital signature certificate of the root CA. If the CA is a root CA, this certificate is its own certificate.

- CRLs

  The CA also adds its CRLs to the processed batch, in order to keep the RA updated in respect to the CRLs. The field is:

  - `crls`: CRLs list.

- Signature

  After adding all the data to the batch, the CA signs it to ensure that the RA will receive the batch without modification of a third party. The only field here is:

  - `casignature`: Batch detached signature generated by the CA.

6.1.5.2 NDCCP messages

NDCCP (Near Domain Cert-status Coverage Protocol) is a Safelayer's proprietary protocol, which is used in the communication between a Database Updater module (in KeyOne VA) and a Cert-status Server module (in KeyOne CA). This communication takes place in order to keep the status database of the KeyOne VA application up-to-date.

The main characteristics of this protocol are:

- Uses HTTPS (HTTP secured with TLS/SSL) as transport mechanism for the messages that are exchanged. Therefore, a trusted channel exists between the Revocation Management Service (KeyOne CA) and the Revocation Status Service (KeyOne VA).

  Therefore, NDCCP messages will be embedded inside the bodies of HTTP request/response messages. In addition, these messages will use the following Content-Type header values:

  - `application/x-safelayer-cert-status-req` for NDCCP request messages.
  - `application/x-safelayer-cert-status-resp` for NDCCP response messages.

- The protocol messages are textually (ASCII) encoded.

The request NDCCP message contains a request identifier, and the related response generated by the KeyOne CertStatus contains also a field indicating the identifier of the associated request. These identifiers act as a nonce, and therefore these requests and responses are protected from replay attacks.

The request contains the time when the message was generated, the signature and the name of the signer (distinguished name of the `subject` field contained in the certificate of the signer).

An NDCCP request message has the following syntax:

```
<sessionID>;<startFrom>;<timeRequest>

[UNSIGNED]

signature = <signature>
```

Where:

`<sessionID>`: Request identifier.

`<startFrom>`: Time reference to consider when selecting the certificates whose status has changed. Only those certificates whose status has changed after this point in time should be returned.

`<timeRequest>`: Time when the request is performed.

`<signature>`: Digital signature of that part of the request message, which precedes the [UNSIGNED] tag.

The NDCCP response message has the following syntax:

<sessionID>;<nextDate>;<timeResponse>; <moreToCome>

<certInfo>

…..

[UNSIGNED]

signature = <signature>

Where:

<sessionID>: Identifier of the associated request.

<nextStartFrom>: Value to include in the <startFrom> field of the next request.

<timeResponse>: Time of response.

<moreToCome>: Flag indicating whether all the certificates matching the request.<startFrom> condition of the request have been included in the response. In case that not, another request with the response.<nextStartFrom> value in its response.<startFrom> field should be issued by the client.

- <certInfo>: Line containing information about a certificate whose status has changed since request.<startFrom>. This line, has the following structure:

  <sn>;<status>;<revreason>;<revdate>;<invdate>;<crtIss>;<holdCode>

  Where:

  - <sn>: Serial number of the certificate.
  - <status>: Current status of the certificate.
  - <revreason>: Revocation reason.
  - <invdate>
  - <certIss>
  - <holdCode>

The message indicates which certificates have been revoked/suspended by means the <sn> field (contained in the <certInfo> structure). The Revocation Status Service (KeyOne VA) requests for certificate status, and the KeyOne CertStatus module queries the KeyOne CA database (status field of the certificates table) and it replies by providing the current status of the certificates included in the NDCCP response (certInfo.status field of the response). Since the KeyOne CertStatus module gets the revocation status from the KeyOne CA database, it provides the current status of the certificates.

6.1.5.3 Functional Requirements satisfied by Security Functions

The Secure Communication services are composed of the following security functions:

- Batch Signature Function (FUNC_BATCHSIG). This functionality generates a signed keyOne batch, providing the integrity, authenticity and non-repudiation security

services to the data contained in the batch. The signature consists of a detached PKCS #7 containing the certification chain (except the root CA certificate).

- Batch Verification Function (FUNC_BATCHVER). This functionality covers the validation by the KeyOne LRA/KeyOne CA of a received KeyOne batch from the KeyOne CA/KeyOne LRA. This validation implies the verification of the digital signature included in the KeyOne batch, and the validation that the author that generated the batch is authorised to do it.

- NDCCP Verification Function (FUNC_NDCCPVER). This functionality covers the validation by the KeyOne VA of a received KeyOne NDCCP message from the KeyOne CA. This validation implies the verification of the digital signature included in the KeyOne message, and the validation that the author that generated the message is authorised to do it.

- NDCCP Signature Function (FUNC_NDCCPSIGCA). This functionality generates a signed NDCCP message in the KeyOne CA component, providing the integrity, authenticity and non-repudiation security services to the data contained in the message. The signature consists of a detached PKCS #7 containing the certification chain (except the root CA certificate).

- Generation of OCSP Responses (FUNC_OCSPRES). This functionality generates an OCSP response in the KeyOne VA component, compliant with the IETF RFC 2560 specifications, after receiving an OCSP request from an OCSP client.

- SSL/TLS between KeyOne Components (FUNC_K1SSLTLS). This functionality is in charge of the establishment of the SSL/TLS protocol between KeyOne components. This secure protocol is used in the communication between the KeyOne LRA and KeyOne CA, and between the KeyOne CA and the KeyOne VA.

- Obfuscation Function (FUNC_OBFUSCATION). This functionality is in charge of protect from unauthorised disclosure and unautorised modifications, the sensitive data managed by the KeyOne system (user data, security settings, administration settings, and others).

These services satisfy the following requirements:

6.1.5.3.1 FDP_ITT.1.1 (iteration 3)

The FDP_ITT.1.1 (iteration 3) requirement needs the integrity service applied to the user data. The user data can be included in the communications between the KeyOne LRA and KeyOne CA components (e.g. register information, …) and in the communication between the KeyOne CA and KeyOne VA (e.g. information about the status of the user certificates).

Regarding to the communication RA-CA, when a certification process is requested, then this request can contain some user data that must be protected against unauthorised modifications. In order to protect this information, after adding all the data to the batch, the KeyOne LRA signs it to ensure that the CA receives the batch without modification of a third-party. The `rasignature` field of the batch contains the batch detached signature generated by the RA. The digital signature generation related to the KeyOne batch is provided by the functionality offered by the FUNC_BATCHSIG function.

Regarding to the communication CA-VA, when the KeyOne CA sends revocation information to the KeyOne VA, then this message contains some user data that must be protected against unauthorised modifications. In order to protect this information, after adding all the data to the NDCCP message, the KeyOne CA signs it to ensure that the KeyOne VA receives the message without modification of a third-party. The signature field of the NDCCP message contains the message detached signature generated by the CA. The digital signature generation related to the KeyOne batch is provided by the functionality offered by the FUNC_NDCCPSIGCA function.

The communications between the KeyOne LRA and the KeyOne CA (KeyOne batch used as data format), and between the KeyOne CA and KeyOne VA (NDCCP message used as data format) use the SSL/TLS secure protocol (with client authentication) in order to provide of the integrity service to these communications. This functionality is provided by the FUNC_K1SSLTLS function.

6.1.5.3.2 FDP_ITT.1.1 (iteration 4)

The FDP_ITT.1.1 (iteration 4) requirement needs the confidentiality service applied to the user data. The user data can be included in the communications between the KeyOne LRA and KeyOne CA components (e.g. register information, …) and in the communication between the KeyOne CA and KeyOne VA (e.g. information about the status of the user certificates).

The communications between the KeyOne LRA and the KeyOne CA (KeyOne batch used as data format), and between the KeyOne CA and KeyOne VA (NDCCP message used as data format) use the SSL/TLS secure protocol (with client authentication) in order to provide of the confidentiality service to these communications. This functionality is provided by the FUNC_K1SSLTLS function.

6.1.5.3.3 FDP_SDI_CIMC.3.1

The public key stored within the CIMC, but not within a FIPS 140-1 validated cryptographic module, are protected against undetected modification through the use of digital signatures.

- If the public key has been certified, then:
  - If the related certificate is a root certificate, because the trusted certificates are stored in a i3D database, then the i3D integrity mechanisms provide the integrity security service.
  - If the related certificate is a non-root certificate, then the integrity service is provided by means the digital signature related to the X.509 certificate.
- If the public key has not been certified, then it can be in the following status:
  - The public key can be contained in the certification request inside the i3D database. In this case, the integrity service is always provided by the integrity applied to the i3D database (the service is provided by the FUNC_I3DSESSION and FUNC_I3DHISTORIC functions, as it is explained in the FDP_SDI_CIMC.3.1 section, page 109).
  - If the request is contained in a KeyOne batch, then because it is signed, then an integrity service is provided to all the information contained in it.

- In the communication between the RA and the CA components, then the integrity applied to the data involved in this communication, is provided by means the digital signature related to the batch, and by means the integrity provided by means the SSL/TLS security protocol. The FUNC_K1SSLTLS function provides the establishment of the SSL/TLS protocol between the KeyOne components.

Therefore, if the public key is included in a KeyOne batch, then the integrity of this key is protected by the digital signature contained in the `rasignature` field of the batch (batch detached signature generated by the RA). The digital signature generation related to the KeyOne batch is provided by the functionality offered by the FUNC_BATCHSIG function.

6.1.5.3.4 FCO_NRO_CIMC.3.1

The FCO_NRO_CIMC.3.1 requirement is accomplished by means the use of functionality offered by the FUNC_BATCHSIG and FUNC_NDCCPSIGCA functions.

This requirement needs the service of generation of evidence of origin for certificate status information and all other security-relevant information at all times.

Because the communication between the KeyOne LRA and KeyOne CA involves information about the certificate status (revocation request from the KeyOne LRA), then this requirement implies an evidence of origin in this communication. In this case, the evidence is provided by means the signature of the KeyOne batch generated by both the KeyOne CA and the KeyOne LRA components. This signature is carried out by means the FUNC_BATCHSIGCA function. The `rasignature` field of the batch contains the batch detached signature generated by the RA, and the `casignature` field of the batch contains the batch detached signature generated by the CA.

KeyOne system is able to relate the identity of the originator of the information and the originator certificate, with the security-relevant portions of the information to which the evidence applies. This evidence consists of the batch signature, that is a detached PKCS #7 containing the certification chain (except the root CA certificate). The identity of the originator of the information is included both in the `rasubject`/`casubject` fields of the batch (author of the generation of the batch), and in the `subject` field of the certificate implied in the batch signature (this certificate is included in the batch). The batches generated by the KeyOne LRA and KeyOne CA components are stored in the batch table of the KeyOne database.

The communication between the KeyOne VA and KeyOne CA also involves information about the certificate status. KeyOne CA sends to the KeyOne VA revocation information about those certificates whose status has changed. his functionality offered by the FUNC_NDCCPSIGCA function consists of the generation of a signed NDCCP message, providing the integrity, authenticity and non-repudiation security services to the data contained in the message. This KeyOne message is used in the communication between the CA (KeyOne CA) and VA (KeyOne VA) components, and the signed response NDCCP message is generated by the Certification Authority.

KeyOne system is able to relate the identity of the originator of the information and the originator certificate, with the security-relevant portions of the information to which the evidence applies. This evidence consists of the message signature, that is a detached PKCS #7 containing. The identity of the originator of the information is

included in the `subject` field of the certificate implied in the NDCCP message signature (this certificate is included in the message).

6.1.5.3.5 FCO_NRO_CIMC.3.2

The FCO_NRO_CIMC.3.2 requirement is accomplished by means the use of functionality offered by the FUNC_BATCHSIG and FUNC_NDCCPSIGCA functions.

This requirement needs the service of generation of evidence of origin for certificate status information and all other security-relevant information at all times.

Because the communication between the KeyOne LRA and KeyOne CA involves information about the certificate status (revocation request from the KeyOne LRA), then this requirement implies an evidence of origin in this communication. In this case, the evidence is provided by means the signature of the KeyOne batch generated by both the KeyOne CA and the KeyOne LRA components. This signature is carried out by means the FUNC_BATCHSIGCA function. The `rasignature` field of the batch contains the batch detached signature generated by the RA, and the `casignature` field of the batch contains the batch detached signature generated by the CA.

KeyOne system is able to relate the identity of the originator of the information and the originator certificate, with the security-relevant portions of the information to which the evidence applies. This evidence consists of the batch signature, that is a detached PKCS #7 containing the certification chain (except the root CA certificate). The identity of the originator of the information is included both in the `rasubject`/`casubject` fields of the batch (author of the generation of the batch), and in the `subject` field of the certificate implied in the batch signature (this certificate is included in the batch). The batches generated by the KeyOne LRA and KeyOne CA components are stored in the batch table of the KeyOne database.

The communication between the KeyOne VA and KeyOne CA also involves information about the certificate status. KeyOne CA sends to the KeyOne VA revocation information about those certificates whose status has changed. his functionality offered by the FUNC_NDCCPSIGCA function consists of the generation of a signed NDCCP message, providing the integrity, authenticity and non-repudiation security services to the data contained in the message. This KeyOne message is used in the communication between the CA (KeyOne CA) and VA (KeyOne VA) components, and the signed response NDCCP message is generated by the Certification Authority.

KeyOne system is able to relate the identity of the originator of the information and the originator certificate, with the security-relevant portions of the information to which the evidence applies. This evidence consists of the message signature, that is a detached PKCS #7 containing. The identity of the originator of the information is included in the `subject` field of the certificate implied in the NDCCP message signature (this certificate is included in the message).

6.1.5.3.6 FCO_NRO_CIMC.3.3

The FCO_NRO_CIMC.3.3 requirement is accomplished by means the use of functionality offered by the FUNC_BATCHVER and FUNC_NDCCPVER functions.

FCO_NRO_CIMC.3.3 requires the verification of the evidence of origin of information for all security-relevant information.

Regarding to the FUNC_BATCHVER function, before processing a certification/revocation request, the KeyOne CA verifies the evidence generated by the RA. If the digital signature verification fails, then an information report and a log registry are generated, and the batch will not be processed. KeyOne CA also verifies that the originator of the evidence (included in the KeyOne batch) is authorised to send certification/revocation requests. KeyOne LRA also verifies the digital signatures contained in the batches received from KeyOne CA, rejecting them when this verification fails.

Regarding to the FUNC_NDCCPVER function, verifies the signature of the NDDCP message. If the digital signature fials, then an information report and a log registry are generated, and the information included in the NDCCP message will not be processed. KeyOne VA also verifies that the originator of the evidence (included in the KeyOne NDCCP message) is authorised to send revocation information, rejecting it when this verification fails.

6.1.5.3.7 FCO_NRO_CIMC.4.1

TheFCO_NRO_CIMC.4.1 requirement is accomplished by means the use of functionality offered by the FUNC_BATCHVER function.

FCO_NRO_CIMC.4.1 requires that the TSF, for initial certificate registration messages sent by the certificate subject, only accept messages protected using an authentication code, keyed hash or digital signature algorithm. Because KeyOne CA only accepts certification batches from a RA that contain digital signatures that can be validated, then FCO_NRO_CIMC.4.1 requirement is accomplished.

6.1.5.3.8 FCO_NRO_CIMC.4.2

The FCO_NRO_CIMC.4.2 requirement is accomplished by means the use of functionality offered by the FUNC_BATCHVER function.

Because KeyOne CA only accepts certification batches from a RA that contain digital signatures that can be validated, then FCO_NRO_CIMC.4.2 requirement is accomplished.

6.1.5.3.9 FDP_CIMC_CSE.1.1

The FDP_CIMC_CSE.1.1 requirement needs that the certificate status information is exported from the KeyOne system in messages whose format complies with the X.509 standard for CRLs and the OCSP standard defined by RFC 2560.

The FUNC_OCSPRES function includes the functionality for generating OCSP responses that will be sent by the KeyOne VA to OCSP clients. This message is generated as a response after receiving and processing the related OCSP request. The OCSP response generated by KeyOne VA is compliant with the format specified in the RFC 2560 (*Online Certificate Status Protocol – OCSP*). This response includes all the mandatory fields of the response, and it is possible to configure some fields and extensions that can be included in the message.

6.1.5.3.10 FDP_CIMC_OCSP.1.1

The FUNC_OCSPRES function includes the functionality for generating OCSP responses that will be sent by the KeyOne VA to OCSP clients. This message is generated as a response after receiving and processing the related OCSP request. The OCSP response generated by KeyOne VA is compliant with the format specified in the RFC 2560 (*Online Certificate Status Protocol – OCSP*). This response includes all the mandatory fields of the response, and it is possible to configure some fields and extensions that can be included in the message.

As the FDP_CIMC_OCSP.1.1 requirement requires, in the OCSP response generation process, the KeyOne VA verifies (FUNC_OCSPRES function) that all mandatory fields in the OCSP basic response contain values in accordance with IETF RFC 2560. Some of the items that will be validate are the following:

- The `version` field shall contain a `0` value.
- If the `issuer` field contains a null Name, then the response will contain a critical `issuerAltName` extension.
- The `signatureAlgorithm` field contains the OID for a FIPS-approved digital signature algorithm.
- The `thisUpdate` field indicates the time at which the status being indicated is known to be correct.
- The `producedAt` field indicates the time at which the OCSP responder signed the response.
- The time specified in the `nextUpdate` field, shall not precede the time specified in the `thisUpdate` field.

6.1.5.3.11 FMT_MOF_CIMC.6.1

The FUNC_OCSPRES function includes the functionality for generating OCSP responses that will be sent by the KeyOne VA to OCSP clients. This message is generated as a response after receiving and processing the related OCSP request. The OCSP response generated by KeyOne VA is compliant with the format specified in the RFC 2560 (*Online Certificate Status Protocol – OCSP*). This response includes all the mandatory fields of the response, and it is possible to configure some fields and extensions that can be included in the message.

The KeyOne VA component implements functionality in order to configure some fields of the OCSP response message.

This functionality is provided by the FUNC_OCSPPROF function. As the FMT_MOF_CIMC.6.1 requires, when the KeyOne VA generates an OCSP response, the consistence of the generated response with respect to the defined OCSP profile is verified.

6.1.5.3.12 FPT_ITC.1.1 (iteration 2)

The FPT_ITC.1.1 (iteration 2) requirement needs the protection against unauthorised disclosure, of the data transmitted from the KeyOne system to a remote trusted IT product.

The sensitive data managed by the KeyOne system (user data, security settings, administration settings, and others) are protected by means the security function FUNC_OBFUSCATION. All these data are stored in a i3D KeyOne Database, and therefore, they shall be transmitted from the KeyOne system to this database. The FUNC_OBFUSCATION function protects the sensitive data used by the KeyOne applications from unauthorised disclosure and unautorised modifications.

6.1.5.3.13 FDP_CIMC_CER.1.3

The FDP_CIMC_CER.1.3 requires that the system verifies that the prospective certificate subject possesses the private key that correspond to the public key in the certificate request before issuing a certificate, unless the public/private key pair was generated by the TSF, whenever the private key may be used to generate digital signatures. In case of private keys that cannot be used to generate digital signatures, then the validation that the requesting subject possesses the private key that correspond to the public key contained in the request, is carried out by means the verification of the KeyOne batch signature generated by an authorised Registration Officer. This verification is included in the functionality provided by the FUNC_BATCHVER function.

6.1.5.3.14 FDP_UCT.1.1 (iteration 2)

This requirement is related to communications of user data and through an external channel (communications between the TOE and a trusted IT product or user).

The communications that are affected with this requirement are the following:

- Communications between the TOE and the Database Server. Because the database client and the KeyOne servers are located in the same host, then the communication between the TOE and the Oracle client is carried out by means physical protection applied to the host. Regarding to the communication between the Oracle client and the Oracle server, it can be protected by means the SSL protocol established between the Oracle client and the Oracle server.

- Communications between the TOE and the SCD (Signature Creation Device), and between the TOE and the HSM (Hardware Security Module) are protected by means physical protection applied to the host.

6.1.5.3.15 FPT_ITT.1.1 (iteration 3)

This requirement requires protection (integrity) to the TSF data, when they are transmitted between separate parts of the TOE.

The FPT_ITT.1.1 (iteration 3) requirement needs the integrity service applied to the TSF data. The TSF data can be included in the communications between the KeyOne LRA and KeyOne CA components, in the communication between KeyOne RA and KeyOne CA and in the communication between the KeyOne CA and KeyOne VA.

In order to protect the TSF data in the communication between KeyOne RA/KeyOne LRA and KeyOne CA, after adding all the data to the batch, the KeyOne RA/KeyOne LRA signs it to ensure that the CA receives the batch without modification of a third-party. The `rasignature` field of the batch contains the batch detached signature generated by the Registration Authority. The digital signature generation related to

the KeyOne batch is provided by the functionality offered by the FUNC_BATCHSIG function.

Regarding to the communication between KeyOne CA and KeyOne VA, when the KeyOne CA sends revocation information to the KeyOne VA, then in order to protect this data, after adding all the data to the NDCCP message, the KeyOne CA signs it to ensure that the KeyOne VA receives the message without modification of a third-party. The signature field of the NDCCP message contains the message detached signature generated by the CA. The digital signature generation related to the KeyOne batch is provided by the functionality offered by the FUNC_NDCCPSIGCA function.

The communications between the KeyOne RA/KeyOne LRA and the KeyOne CA (KeyOne batch used as data format), and between the KeyOne CA and KeyOne VA (NDCCP message used as data format) use the SSL/TLS secure protocol (with client authentication) in order to provide of the integrity service to these communications. This functionality is provided by the FUNC_K1SSLTLS function.

6.1.5.3.16 FPT_ITT.1.1 (iteration 4)

This requirement requires protection (confidentiality) to the TSF data, when they are transmitted between separate parts of the TOE.

The FPT_ITT.1.1 (iteration 4) requirement needs the confidentiality service applied to the TSF data. The TSF data can be included in the communications between the KeyOne RA/KeyOne LRA and KeyOne CA components and in the communication between the KeyOne CA and KeyOne VA.

The communications between the KeyOne RA/KeyOne LRA and the KeyOne CA (KeyOne batch used as data format), and between the KeyOne CA and KeyOne VA (NDCCP message used as data format) use the SSL/TLS secure protocol (with client authentication) in order to provide of the confidentiality service to these communications. This functionality is provided by the FUNC_K1SSLTLS function.

6.1.5.3.17 FDP_CIMC_BKP.2.1

The FDP_CIMC_BKP.2.1 requirement needs that the backup data (generated by the FUNC_BACKUP security function) shall be protected against modification through the use of digital signatures, keyed hashes, or authentication codes.

The KeyOne System includes a functionality of backup that is in charge of backing up the whole KeyOne system necessary to reconstruct the current status from this backup and a copy of the same version of the software used to install initially the KeyOne system. The data stored in the system backup necessary to recreate the state of the system at the time the backup includes all the necessary information stored in the hard disc of the machine where the KeyOne system is installed. This information is protected by means the FUNC_OBFUSCATION security function that protects these data from unautorised modifications.

6.1.5.3.18 FDP_CIMC_BKP.2.2

The FDP_CIMC_BKP.2.2 requirement needs that the critical parameters and other confidential information of the backup data (generated by the FUNC_BACKUP security function) shall be stored in encrypted form only.

The KeyOne System includes a functionality of backup that is in charge of backing up the whole KeyOne system necessary to reconstruct the current status from this backup and a copy of the same version of the software used to install initially the KeyOne system. The data stored in the system backup necessary to recreate the state of the system at the time the backup includes all the necessary information stored in the hard disc of the machine where the KeyOne system is installed, and all the information stored in the KeyOne Databases. This information is protected by means the FUNC_OBFUSCATION security function that protects these data from unautorised disclosure by means a encryption process.

6.1.6 Certification Management

An important part of the KeyOne system is all the management of certification issues: verification of certification requests, generation of certificates and CRLs, and management of certification profiles. This section contains information about the requirements and functions that are related to these security aspects.

6.1.6.1 Functional Requirements satisfied by Security Functions

The Certification Management services are composed of the following security functions:

- Certification Request Verification Function (FUNC_CERTREQVER). This functionality is in charge of the verification of the certification request received by the KeyOne CA. This validation includes the verification of the Proof Of Possession included in the certification request and generated by the Registration Authority.
- Certificates Generation Function (FUNC_CERTSGENE). This functionality is in charge of the generation of certificates following the X.509 standard for public key certificates. The function assures that the generated certificates are consistent with the currently defined certificate profile.
- PKCS #12 Generation Function (FUNC_PKCS12GENE). This functionality is in charge of the generation of PKCS #12 following the specifications of the "PKCS #12 Personal Information Exchange Syntax".
- CRLs Generation Function (FUNC_CRLSGENE). This functionality is in charge of the generation of CRLs in accordance with the ITU-T Recommendation X.509. The function assures that the generated certificates are consistent with the currently defined certificate profile.
- Certification Profile (FUNC_CERTPROF). This functionality is in charge of providing to the KeyOne CA of the functionality in order to manage certification profiles by an authorised Administrator.
- Revocation Profile (FUNC_REVPROF). This functionality is in charge of providing to the KeyOne CA of the functionality in order to manage revocation profiles by an authorised Administrator.
- OCSP Profile (FUNC_OCSPPROF). This functionality is in charge of providing to the KeyOne VA of the functionality in order to manage OCSP profiles by an authorised Administrator.

These services satisfy the following requirements:

6.1.6.1.1 FDP_CIMC_CER.1.3

The FDP_CIMC_CER.1.3 requires that the system verifies that the prospective certificate subject possesses the private key that correspond to the public key in the certificate request before issuing a certificate, unless the public/private key pair was generated by the TSF, whenever the private key may be used to generate digital signatures.

In case of the private keys could be used to generate digital signatures, then the validation that the requesting subject possesses the private key that correspond to the public key contained in the request, is carried out by means the verification of the signature generated by the requesting entity that is included in the PKCS #10 or X.509 certification request. The FUNC_CERTREQVER function verifies the self-signed certification request that is included in a certification KeyOne batch.

6.1.6.1.2 FDP_CIMC_CER.1.1

The Certificates Generation Function (FUNC_CERTSGENE) is in charge of the generation of certificates following the X.509 standard for public key certificates, in the KeyOne CA component. Therefore the FDP_CIMC_CER.1.1 requirement is accomplished.

6.1.6.1.3 FDP_CIMC_CER.1.2

Before the generation of the X.509 certificate, the FUNC_CERTSGENE function assures that the generated certificates are consistent with the currently defined certificate profile. Consequently the FUNC_CERTSGENE function covers the accomplishment of the FDP_CIMC_CER.1.2 requirement.

6.1.6.1.4 FDP_SDI_CIMC.3.1

The FDP_SDI_CIMC.3.1 requirement forces to provide of the integrity service (by means digital signatures, keyed hashes or authentication codes) to the public keys stored within the CIMC but not within a FIPS 140-1 validated cryptographic module. In case of the public key has already been certified, then the integrity of it is supplied by means the digital signature related to the certificate. Because the X.509 standard includes the signature field in is format, then the FUNC_CERTSGENE function covers the accomplishment of the FDP_SDI_CIMC.3.1 requirement.

6.1.6.1.5 FMT_MOF_CIMC.3.1

Before the generation of the X.509 certificate, the FUNC_CERTSGENE function assures that the generated certificates are consistent with the currently defined certificate profile. Consequently the FUNC_CERTSGENE function covers the accomplishment of the FMT_MOF_CIMC.3.1 requirements.

6.1.6.1.6 FDP_CIMC_CER.1.4

The FUNC_CERTSGENE function checks the following restrictions in the generation of X.509 certificates:

- The `version` field shall contain the integer `0`, `1` or `2`.

- If the certificate contains an `issuerUniqueID` or `subjectUniqueID` then the version field shall contain the integer `1` or `2`.
- If the certificate contains `extensions` then the `version` field shall contain the integer `2`.
- The `serialNumber` shall be unique with respect to the issuing Certification Authority.
- The validity field shall specify a `notBefore` value that does not precede the current time and a `notAfter` value that does not precede the value specified in `notBefore`.
- If the `issuer` field contains a null Name, then the certificate shall contain a critical `issuerAltName` extension.
- If the `subject` field contains a null Name, then the certificate shall contain a critical `subjectAltName` extension.
- The `signature` field and the algorithm in the `subjectPublicKeyInfo` field shall contain the OID for a FIPS-approved or recommended algorithm.

Because the FUNC_CERTSGENE function checks these restrictions, then this function covers the accomplishment of the FDP_CIMC_CER.1.4 requirement.

6.1.6.1.7 FDP_ETC_CIMC.5.1

The PKCS# 12 Generation Function (FUNC_PKCS12GENE) is in charge of the generation of PKCS #12 structures following the specifications of the "PKCS #12 Personal Information Exchange Syntax" in the KeyOne CA component.

The FDP_ETC_CIMC.5.1 requirement forces to the exportation of private and secret keys from the TOE, in encrypted form or using split knowledge procedures. In case of electronic distribution of the secret and private keys, then they shall only be exported from the TOE in encrypted form.

Regarding to the exportation of private keys in a PKCS #12 format, then because the FUNC_PKCS12GENE function follows the PKCS #12 specifications, then this requirement is accomplished (the PKCS #12 specification is based on a privacy mode that use the encryption to protect personal information from exposure).

6.1.6.1.8 FDP_CIMC_CRL.1.1

The CRLs Generation Function (FUNC_CRLSGENE) is in charge of the generation of CRLs in accordance with ITU-T Recommendation X.509, in the KeyOne CA component. This function checks the following restrictions in the generation of CRLs:

- If the `version` field is present, then it shall contain a `1`.
- If the CRL contains any critical `extensions` then the `version` field shall contain the integer 1.
- If the `issuer` field contains a null Name, then the CRL shall contain a critical `issuerAltName` extension.

- The `signature` field and the `signatureAlgorithm` fields shall contain the OID for a FIPS-approved or recommended algorithm.
- The `thisUpdate` field shall indicate the issue data of the CRL.
- The time specified in the `nextUpdate` field shall not precede the time specified in the `thisUpdate` field.

Therefore the FDP_CIMC_CRL.1.1 requirement is accomplished.

6.1.6.1.9 FMT_MOF_CIMC.5.1

The CRLs Generation Function (FUNC_CRLSGENE) is in charge of the generation of CRLs in accordance with ITU-T Recommendation X.509, in the KeyOne CA component.

Before the generation of the X.509 CRL, the FUNC_CRLSGENE function assures that the issued CRL is consistent with the certificate revocation list profile. Consequently the FUNC_CRLSGENE function covers the accomplishment of the FMT_MOF_CIMC.5.1 requirement.

6.1.6.1.10 FMT_MOF_CIMC.3.1

This requirement implies the functionality that allows managing a certificate profile by the authorised Administrator. Through the Certification Profile Function (FUNC_CERTPROF) KeyOne CA provides of a certification template mechanism.

Certification templates determine features of the certificates issued by the CA (like the certificate extensions). A certification template, also called certification policy or simply policy, is a set of programmable rules that define constraints on a certain type of certificate requests that the CA will accept, as well as the characteristics of the certificates issued from that type of requests (like the certificate extensions).

Multiple certification templates may be defined for the CA, one for each type of certificate request that is to be processed. Requests may be classified in different types according to the intended certificate uses, the type of entity for which the certificate is issued or any other criteria. The set of certification templates does not need to be exhaustive, that is, it is not necessary to define certification templates for every possible request type. Moreover, many templates may be defined for the same request type with some differences like the certificate validity period. The authorised CA Administrator must give a name to each certification template when defining it.

Alternatively, when a single certificate or PKCS #12 is directly issued from the KeyOne CA administration application, the CA administrator must explicitly select the certification template to be applied.

Certification Template Application

To issue a certificate, KeyOne CA applies a certification template to a certain certificate request. This is known as certification template application and it is the first step of the certificate issuing process. This step consists of applying the rules defined in the certification template for the various certificate fields and extensions, taking into account the values proposed in the certificate request in some cases. This may result in fields and extensions being added, removed or modified.

After the certification template application, a second step is needed to complete the certificate issuing process. This step is called certificate/PKCS #12 generation and it consists of signing the certificate and possibly generating a PKCS #12, if the original request did not include a public key (the PKCS #12 will be used to deliver the certificate along with its associated private key to the certificate owner).

The certificate/PKCS #12 generation step will not be performed if the certification template cannot be applied to the certificate request.

Certification vs. Certificate Templates

As the result of applying a certification template to a certificate request, a certificate template is obtained. A certificate template contains the same information that a certificate but it does not include a signature (a certificate that is not yet signed). In fact, the original certificate request may also be represented as a certificate template. The internal representation of a certificate template is the `CertTemplate` ASN.1 structure defined in IETF RFC 2511.

Besides fields and extensions that will be part of the final certificate, the certificate template may also include other information that will not be directly included in the certificate or not even used to build the certificate itself. In particular, when the original certificate request does not include a public key, the certificate template resulting from the certification template application will include information on:

- How the key pair is to be generated by the CA engine in the later certificate generation phase (this includes information on the key algorithm and related parameters according to the particular algorithm, e.g. the RSA key size).
- The password of the PKCS #12 to generate along with the certificate. Usually this information is also included in the original certificate request.

Certification Template Rules

The definition of a certification template consists of various fields, each one determining how a certain certificate field or extension must be set for certificates that will be issued with that template. Examples of certificate fields are the certificate version and the validity period. Examples of extensions include the subject alternative names or the basic constraints extension defined in the X.509 standard (from now on, the term field will be used to refer to either certificate fields or certificate extensions).

For some fields, the certificate request may propose a value for the field to be included in the issued certificate. In these cases, constraints on the values that are to be allowed may be imposed in the certification template. These are called negotiable fields.

For each field in the certification template a set of application rules may be defined. These rules are of the following types:

- Field absence or presence

  Determines whether the field should be included or not in the issued certificates.

  Such rules allow controlling what extensions will be included.

- Field optionality

For some negotiable fields it is possible to specify that the field should be included only if the certificate request contains a value for it.

- Field value

  Such rules determine the value the field must take in the issued certificate.

- Field value constraints

  For some negotiable fields it is possible to specify constraints or ranges that the value indicated in the certificate request for that field or extension must satisfy in order to be included in the issued certificate. This allows limiting values that can be requested.

- Default field value

  When a negotiable field is specified to be always present, this rule determines the value the field must take when it is not included in the certificate request.

- Extension criticality

  Such rules determine whether a specific extension should be marked as critical or not when included in the issued certificate.

Any extension contained in the certificate request not corresponding to any field in the certification template will be ignored and it will not be included in the issued certificate.

Templates may be added, imported, examined, modified and removed at any moment. KeyOne CA requires that at least one certification template be defined before starting to issue certificates.

6.1.6.1.11 FMT_MOF_CIMC.3.2

These requirements imply the functionality that allows managing a certificate profile by the authorised Administrator. Through the Certification Profile Function (FUNC_CERTPROF) KeyOne CA provides of a certification template mechanism.

For more information about certification templates or the FUNC_CERTPROF function, see FMT_MOF_CIMC.3.1, page 152.

6.1.6.1.12 FMT_MOF_CIMC.3.3

These requirements imply the functionality that allows managing a certificate profile by the authorised Administrator. Through the Certification Profile Function (FUNC_CERTPROF) KeyOne CA provides of a certification template mechanism.

For more information about certification templates or the FUNC_CERTPROF function, see FMT_MOF_CIMC.3.1, page 152.

6.1.6.1.13 FMT_MOF_CIMC.3.4

These requirements imply the functionality that allows managing a certificate profile by the authorised Administrator. Through the Certification Profile Function (FUNC_CERTPROF) KeyOne CA provides of a certification template mechanism.

For more information about certification templates or the FUNC_CERTPROF function, see FMT_MOF_CIMC.3.1, page 152.

6.1.6.1.14 FMT_MOF_CIMC.5.1

This requirement implies the functionality that allows managing a revocation profile by the authorised Administrator. Through the Revocation Profile Function (FUNC_REVPROF) KeyOne CA provides of a revocation template mechanism.

Whereas a certification template defines fields and extensions for some type of issued certificates, a CRL template determines how KeyOne CA must set fields and extensions for a particular Certificate Revocation List (CRL) issued by the CA. Examples of CRL fields are the CRL version and the CRL next updating date. Examples of CRL extensions are the CRL number and the issuing distribution point extensions defined in the X.509 standard.

Unlike a certification template, a CRL template is not applied to any request. Instead, the CRL template is directly used to generate a CRL, along with information stored in the CA database about the current revoked certificates. Because of this difference, we do not talk of CRL template application rules but simply of CRL template fields.

The CRL Template Set

The CA may issue only one CRL or several of them. In the latter case, each CRL will commonly cover a distinct set of revocation reasons or entity types, and the various CRLs will be assigned different CRL distribution points (methods of obtaining CRL information). This information is also defined in the corresponding CRL templates.

A CRL template must be defined for each CRL the CA should issue. At least one CRL template must be defined. Unlike with certification templates, since the number of CRL templates determines the number of CRLs this parameter is not expected to vary during the CA's life. Furthermore, CRL templates should be fully defined before starting to issue certificates so that information on the CRL distribution points may be properly included in certificates.

Generation of CRLs using the CRL templates

Whenever the CA CRLs must be issued, KeyOne CA will use the defined CRL template set to generate the CRLs. This process may involve all CRL templates or only some of them, depending on which CRLs should be updated. For instance, the first time the CRLs are generated all CRL templates are used. On the contrary, if CRLs are being updated then only those CRL that have expired or whose contents must change are generated again, for which the corresponding CRL templates are used.

When the CRL corresponding to a certain CRL template must be issued (either for the first time or because it needs to be updated), the CRL template is used to determine the following:

- The value of some CRL fields (for instance the CRL version and the CRL next-update date). Fields not specified in the CRL template are automatically set by KeyOne CA.
- The extensions the CRL should include and whether they are critical or not. For some extensions, the extension value may also be defined by the CRL template

(e.g. the issuing distribution point extension). In other cases, the extension value is automatically calculated by KeyOne CA (e.g. the CRL number extension).

- Which revoked certificates must be included in the CRL. This applies only if revocation reasons to be covered by the CRL have been specified in the CRL template. In this case, KeyOne CA will automatically include each revoked certificate in the appropriate CRL(s) according to the certificate revocation reason (this information and other certificate data is obtained from the CA database).

Furthermore, as mentioned above, the defined CRL template set is not only used when issuing CRLs but also when issuing certificates. Concretely, CRL templates are used to determine whether the CRL distribution points extension should be included in each issued certificate and its value.

6.1.6.1.15 FMT_MOF_CIMC.5.2

This requirement implies the functionality that allows managing a revocation profile by the authorised Administrator. Through the Revocation Profile Function (FUNC_REVPROF) KeyOne CA provides of a revocation template mechanism.

For more information about CRL templates or the FUNC_REVPROF function, see FMT_MOF_CIMC.5.1, page 155.

6.1.6.1.16 FMT_MOF_CIMC.5.3

This requirement implies the functionality that allows managing a revocation profile by the authorised Administrator. Through the Revocation Profile Function (FUNC_REVPROF) KeyOne CA provides of a revocation template mechanism.

6.1.6.1.17 FMT_MOF_CIMC.6.1

This requirement implies the functionality that allows managing a profile for the OCSP responses, by the authorised Administrator. Through the OCSP Profile Function (FUNC_OCSPROF) KeyOne VA provides the functionality in order to configure some fields of the OCSP response message.

KeyOne VA allows generating basic OCSP responses, and an authorised Administrator can configure certain fields of this type of response.

6.1.6.1.18 FMT_MOF_CIMC.6.2

This requirement implies the functionality that allows managing a profile for the OCSP responses, by the authorised Administrator. Through the OCSP Profile Function (FUNC_OCSPROF) KeyOne VA provides the functionality in order to configure some fields of the OCSP response message.

KeyOne VA allows generating basic OCSP responses, and an authorised Administrator can configure certain fields of this type of response.

6.1.6.1.19 FMT_MOF_CIMC.6.3

This requirement implies the functionality that allows managing a profile for the OCSP responses, by the authorised Administrator. Through the OCSP Profile Function (FUNC_OCSPROF) KeyOne VA provides the functionality in order to configure some fields of the OCSP response message.

KeyOne VA allows generating basic OCSP responses, and an authorised Administrator can configure certain fields of this type of response.

6.1.6.1.20 FDP_ACF_CIMC.2.1

The CIMC personnel private keys are stored in a FIPS 140-2 level 2.

6.1.6.1.21 FDP_ACF_CIMC.2.2

The only certificate subject private keys that are stored in the TOE are the keys that are been copied by the KeyOne Archive Component (for Key Recovery purposes).

These keys are encrypted using Long Term Private Key Protection Key (Component Keys), and this encryption is performed by a cryptographic module, that is a FIPS 140-2 level 2 validated cryptographic module (Administrator of the KeyArchive component).

6.1.6.1.22 FDP_ACF_CIMC.3.1

The only user keys that are stored within the CIMC but not within a FIPS validated cryptographic module, are the subject private keys.

These keys are the keys that are been copied by the KeyOne Archive Component (for Key Recovery purposes). These keys are encrypted using a cryptographic module (Administrator of the Key Archive Component) that is a FIPS 140-2 level 2 validated cryptographic module.

6.1.6.1.23 FMT_MTD_CIMC.4.1

The CIMC private keys are stored in a FIPS 140-2 level 3 validated cryptographic module. When these keys are exported to the TOE (shutdown of the system), then they are ciphered by this FIPS 140-2 level 3 Hardware Security Module.

6.1.6.1.24 FMT_MTD_CIMC.5.1

The FMT_MTD_CIMC.5.1 requires that the TSF secret keys stored within the TOE, but not within a FIPS 140-1 validated cryptographic module, be stored in encrypted form. This encryption shall be performed by the FIPS 140-1 validated cryptographic module.

All the secret keys are either stored in FIPS 140-2 level 3 validated cryptographic modules, or they are ciphered using FIPS 140-2 level 3 Hardware Security Module.

6.1.6.1.25 FMT_MTD_CIMC.7.1

The private and secret keys are not exported from the TOE. The private and secret keys are maintained in the secure store where they were generated, and never can be exported from there.

6.1.6.1.26 FCS_CKM_CIMC.5.1

The TOE does not maintain plaintext secret and private keys.

6.1.7 Private Secure Store

The Private Secure Store is a safe object where sensitive configuration data is stored and protected against illicit access or modification. Examples of information that can be stored in this protected store are: private keys, root certificates, configuration data, etc.

This KeyOne Private Secure Store typically stores the following data of the KeyOne applications:

- Configuration data of the system
- Configuration data of the applications
- Service keys of the applications. Service keys (application) are considered the set of asymmetric keys that an KeyOne 3.0 application needs in order to correctly operate: infrastructure keys (SSL, signature of KeyOne batches, …), key for signing certificates and CRLs, key for signing OCSP messages, …
- Data Obfuscation keys. Data Obfuscation is a generic term that includes all the auxiliary keys (symmetric or asymmetric) that are implied in the data protection mechanisms of KeyOne 3.0 (master keys, key keys, data keys, …).

The Private Secure Store allows storing data in tree format, identifying each entry by a type, a name and the superior entry (father entry). For each entry, it is possible to define a set of attributes, each one of them having a name and a value. It is possible to define attributes that are references to other entries, and attributes that are external references (references to entries of other Private Secure Stores).

The implementation of the registry uses two i3D tables, but the following data will be stored in the disc (necessary data in order to access to the Private Secure Store):

- The configuration of the data obfuscation key.
- The data obfuscation keys.
- The key that protects the configuration of the system database.
- The configuration of the system database.
- Service keys of the applications.
- System certificates (application certificates and certificates that are needed to operate).

6.1.7.1 Private Secure Store Functionality

The main functionality of the Private Secure Store is to be a secure store for certificates, keys and configuration data. The Private Secure Store also keeps a historic record of expired certificates. All the data stored in the Private Secure Store is protected from unauthorized access and modification using cryptographic techniques.

The Private Secure Store can store any kind of signed objects, like certificates. The signature of any signed object is validated before it is inserted in the Private Secure Store. If the object's signature cannot be validated, the object is not inserted. Following this rule, it is easy to see that the Private Secure Store stores complete certificate hierarchies.

Signed objects (like certificates) are not valid forever. Each object has an expiration date. When the expiration date is reached, the object cannot be used anymore and must be deleted from the Private Secure Store. The mechanism to delete expired objects is the following:

- On every access for a certain object, the Private Secure Store goes around several objects until it finds the requested object. Any invalid object that the Private Secure Store finds during this search is deleted.
- Not all Private Secure Store objects are validated when searching a certain object. There can be invalid objects remaining in the Private Secure Store, but these objects will be deleted the first time anyone tries to access them.
- Expired certificates whose private key is stored in the Private Secure Store are not deleted, but moved to the Private Secure Store historic record.

To protect the Private Secure Store from unauthorised modifications (integrity service), the two following mechanisms are used:

- Integrity mechanism used in the i3d database (applied over the data included in the i3D table).
- Integrity mechanism used in the Private Secure Store. This mechanism consists of the application of a hash algorithm to the data to be protected. The result of the hash application is ciphered with the other data using a symmetric algorithm. Both the type of the hash algorithm and the symmetric ciphering algorithm are configurable.

The Private Secure Store also is protected against unauthorised access (confidentiality service) by means the ciphering of the data using the data obfuscation mechanism (the type of the algorithm is configurable).

6.1.7.1.1 Historic record

The Private Secure Store historic record stores the expired certificates and its corresponding private keys. When a certificate expires and the Private Secure Store notices it, the existence of its private key in the Private Secure Store is checked. If the private key is present, the certificate and its private key are moved into the historic record. Otherwise, the certificate is deleted.

The expired certificates with private keys are kept in the historic record for deciphering encrypted data and/or the need to validate some signatures done with the private key when the certificate was still valid. If the certificate is deleted and not kept in the historic record, the data will remain encrypted forevermore because the signatures will fail to be validated.

6.1.7.2 Functional Requirements satisfied by Security Functions

The Private Secure Store Management services is composed of the following security function:

- Private Secure Store Access Function (FUNC_PSSINSERT). This functionality is in charge of the verification of the signed objects, when they are inserted in the Private Secure Store. The function will delete the invalid objects found (expired certificates whose private key is stored in the Private Secure Store are not deleted but moved to the Private Secure Store historic record).

This service satisfies the following requirements:

6.1.7.2.1 FDP_SDI_CIMC.3.2

The FDP_SDI_CIMC.3.2 requires that the digital signature, keyed hash or authentication code used to protect a public key be verified upon each access to the key.

The public keys are protected by using the digital signature related to the certificate. These certificates are stored in the Private Secure Store (PSS), and the integrity of the public keys stored in the PSS of the following way:

- The FUNC_PSSINSERT function guarantees that each time that a new signed object (certificate, CRL) is inserted in the Private Secure Store, then it is validated (the digital signature related to the certificate or CRL is also validated).
- The content of the Private Secure Store is maintained in the Registry I3D Database, and therefore the integrity of the data kept in this database is assured by the FUNC_I3DSESSION and FUNC_I3DHISTORIC functions.
- For each access to any object stored in the PSS, the expiration data is checked. Expired objects are deleted from the Private Secure Store, and it use is forbidden.
- When any application starts, then the content of the Private Secure Stored is read from the I3D database in order to copy it into the machine memory. In this step the integrity of the PSS is verified, and therefore the integrity of the public keys is assured.

6.1.8 Key Archive Management

The TOE includes the KeyOne Archive component in order to securely store keys generated by the KeyOne CA. The keys are stored in a secure database in PKCS #12 format (this PKCS #12 is symmetrically ciphered by a symmetric key maintained in the PKCS #11 device). KeyOne Archive incorporates the additional role Key Recovery Officer. The Key Archive component can be configured in order to the recovery process requires N Key Recovery Officers (N>=1).

There ca be N separated officers for security reasons, so that the same person cannot access the archived keys. The connection between the Key Recovery Officers and the key recovery storage is carried out in a remote way through a secure web connection. Key Recovery Officers have remote access to the database where the PKCS #12 are stored.

If someone has lost their private key or does not remember the access password to the PKCS #12, the Administrators can supply them, if they have previously proven their identity (the Key Recovery Officer must be added into the KeyOne Console as a Key Recovery Officer user). Once the Key Recovery Officers has been authenticated, then the PKCS #12 will be generated and it be delivered to the end user. The PIN related to this PKCS #12 also will be generated and it will be split in N pieces (N number of Key Recovery Officers). Each Key Recovery Officer will have a piece of the PIN, and it will be delivered to the end user, for building the original PIN related to the PKCS #12.

6.1.8.1 Functional Requirements satisfied by Security Functions

The Key Archive Management services are composed of the following security functions:

- Key Recovery Function (FUNC_KEYRECOV). This functionality is in charge of the recovery of private keys by means the Key Recovery Officers. This function is located in the KeyOne Key Archive component (KeyOne CA product).

These services satisfy the following requirements:

6.1.8.1.1 FDP_ETC_CIMC.5.1

The FDP_ETC_CIMC.5.1 requires that the private and secret keys only could be exported from the TOE in encrypted form or using split knowledge procedures. In case of electronic distribution, these keys only can be exported from the TOE in encrypted form.

Regarding to the exportation of private keys in a Key Recovery process, this exportation accomplishes with the constraints included in the FDP_ETC_CIMC.5.1 requirement. Because the Key Recovery Function exports the private key in a symmetrically ciphered PKCS #12 format, the private key is exported in encrypted form.

6.1.9 Backup and Recovery

The TOE includes the Backup and Recovery functionality that is in charge of reconstructing a system in the event of a system failure or other serious error. In order to be able to recover from failures and other unanticipated undesired events, the KeyOne system is able to back up the system. The KeyOne backup will be used to restore the KeyOne system to an operational status at a previous point in time.

This functionality only can be invoked by the System Administrator role, and only this role can configure the parameters involved in this functionality.

The data stored in the system backup necessary to recreate the state of the system at the time the backup are the following:

- Cryptographic keys and other data stored in the HSM used.
- All the information stored in the KeyOne Databases.
- All the necesssary information stored in the hard disc of the machine where the KeyOne system is installed.

6.1.9.1 Functional Requirements satisfied by Security Functions

The Backup and Recovery services are composed of the following security functions:

- Backup and Recovery Function (FUNC_BACKUP). This functionality involves tasks related to the Backup and Recovery functionality located in the KeyOne system.

These services satisfy the following requirements:

6.1.9.1.1 FDP_CIMC_BKP.1.1

The FDP_CIMC_BKP.1.1 requirement requires that the TOE provides a backup function. The FUNC_BACKUP security function implements a command-line tool that is related to the backup process. This backup process backus up the whole KeyOne system necessary to reconstruct the current status from this backup and a copy of the same version of the distributin and patches used to install initially the KeyOne system.

6.1.9.1.2 FDP_CIMC_BKP.1.2

The FDP_CIMC_BKP.1.2 requirements requires the possibility to invoke the backup function on demand. The FUNC_BACKUP security function implements a command-line tool that is related to the backup process. This backup process backus up the whole KeyOne system necessary to reconstruct the current status from this backup and a copy of the same version of the distributin and patches used to install initially the KeyOne system. The System Administrator role is able to invoke the backup process on demand.

6.1.9.1.3 FDP_CIMC_BKP.1.3

The FDP_CIMC_BKP.1.3 is assured by means the FUNC_BACKUP security function. The FUNC_BACKUP security function implements a command-line tool that is related to the backup process. The data stored in the system backup necessary to recreate the state of the system at the time the backup are the following:

- Cryptographic keys and other data stored in the HSM used.
- All the information stored in the KeyOne Databases.
- All the necessary information stored in the hard disc of the machine where the KeyOne system is installed.

6.1.9.1.4 FDP_CIMC_BKP.1.4

The FDP_CIMC_BKP.1.4 requirement requires that the TOE provides a recovery function that is able to restore the state of the system from a backup. The FUNC_BACKUP security function implements a command-line tool that is related to the recovery process. This restore process reconstruct the status of the KeyOne system from the result of the backup process and a copy of the same version of the distribution and patches used to install initially the KeyOne system.

6.2 Mapping Table between functional requirements and security functions

This section includes a mapping table between the TOE security functional requirements included in this Security Target and the TOE security functions specified in the [FUNCSPEC] document.

Additionally, a mapping between TOE security functions and TOE security functional requirements has been included.

Functional Requirement	*Security Function*
FAU_GEN.1.1 (Iter. 2)	FUNC_SADG
FAU_GEN.1.2 (Iter. 2)	FUNC_SADG
FAU_GEN.2.1 (Iter. 2)	FUNC_SADG
FAU_SEL.1.1 (Iter. 2)	FUNC_SELL
FAU_STG.1.1 (Iter. 2)	(The TSF does not have any functionality to delete registries from the audit database)
FAU_STG.1.2 (Iter. 2)	FUNC_DBIV
FAU_STG.4.1 (Iter. 2)	FUNC_CDBC
FPT_STM.1.1 (Iter. 2)	FUNC_I3DSESSION, FUNC_I3DHISTORIC
FMT_MOF.1.1 (Iter. 2)	FUNC_ACCESSCTRL
FDP_ACC.1.1 (Iter. 2)	FUNC_ACCESSCTRL
FDP_ACF.1.1 (Iter. 2)	FUNC_ACCESSCTRL
FDP_ACF.1.2 (Iter. 2)	FUNC_ACCESSCTRL
FDP_ITT.1.1 (Iter. 3)	FUNC_BATCHSIG, FUNC_NDCCPSIGCA, FUNC_K1SSLTLS
FDP_ITT.1.1 (Iter. 4)	FUNC_K1SSLTLS
FDP_UCT.1.1 (Iter. 2)	See Note about FDP_UCT.1.1 requirement in this section
FPT_RVM.1.1 (Iter. 2)	FUNC_ACCESSCTRL

FPT_ITC.1.1 (Iter. 2)	FUNC_OBFUSCATION
FIA_UAU.1.1 (Iter. 2)	FUNC_UIDAUT
FIA_UAU.1.2 (Iter. 2)	FUNC_UIDAUT
FIA_UID.1.1 (Iter. 2)	FUNC_UIDAUT
FIA_UID.1.2 (Iter. 2)	FUNC_UIDAUT
FIA_USB.1.1 (Iter. 2)	FUNC_UIDAUT
FPT_ITT.1.1 (Iter. 3)	FUNC_BATCHSIG, FUNC_NDCCPSIGCA, FUNC_K1SSLTLS
FPT_ITT.1.1 (Iter. 4)	FUNC_K1SSLTLS
FPT_CIMC_TSP.1.1	FUNC_I3DSESSION
FPT_CIMC_TSP.1.2	FUNC_I3DSESSION
FPT_CIMC_TSP.1.3	FUNC_I3DSESSION, FUNC_I3DHISTORIC
FPT_CIMC_TSP.1.4	FUNC_I3DSESSION
FDP_SDI_CIMC.3.1	FUNC_I3DSESSION, FUNC_I3DHISTORIC, FUNC_BATCHSIG, FUNC_K1SSLTLS, FUNC_CERTSGENE
FDP_SDI_CIMC.3.2	FUNC_PSSINSERT, FUNC_I3DSESSION, FUNC_I3DHISTORIC
FDP_ETC_CIMC.5.1	FUNC_PKCS12GENE, FUNC_KEYRECOV
FDP_CIMC_CSE.1.1	FUNC_OCSPRES
FDP_CIMC_CER.1.1	FUNC_CERTSGENE
FDP_CIMC_CER.1.2	FUNC_CERTSGENE
FDP_CIMC_CER.1.3	FUNC_BATCHVER, FUNC_CERTREQVER
FDP_CIMC_CER.1.4	FUNC_CERTSGENE
FDP_CIMC_CRL.1.1	FUNC_CRLSGENE
FDP_CIMC_OCSP.1.1	FUNC_OCSPRES
FCO_NRO_CIMC.3.1	FUNC_BATCHSIG, FUNC_NDCCPSIGCA
FCO_NRO_CIMC.3.2	FUNC_BATCHSIG, FUNC_NDCCPSIGCA
FCO_NRO_CIMC.3.3	FUNC_BATCHVER, FUNC_NDCCPVER
FCO_NRO_CIMC.4.1	FUNC_BATCHVER
FCO_NRO_CIMC.4.2	FUNC_BATCHVER
FMT_MTD_CIMC.7.1	(The private and secret keys are not exported from the TOE)
FMT_MOF_CIMC.3.1	FUNC_CERTSGENE, FUNC_CERTPROF
FMT_MOF_CIMC.3.2	FUNC_ACCESSCTRL, FUNC_CERTPROF

FMT_MOF_CIMC.3.3	FUNC_ACCESSCTRL, FUNC_CERTPROF
FMT_MOF_CIMC.3.4	FUNC_ACCESSCTRL, FUNC_CERTPROF
FMT_MOF_CIMC.5.1	FUNC_CRLSGENE, FUNC_REVPROF
FMT_MOF_CIMC.5.2	FUNC_ACCESSCTRL, FUNC_REVPROF
FMT_MOF_CIMC.5.3	FUNC_ACCESSCTRL, FUNC_REVPROF
FMT_MOF_CIMC.6.1	FUNC_OCSPRES, FUNC_OCSPPROF
FMT_MOF_CIMC.6.2	FUNC_ACCESSCTRL, FUNC_OCSPPROF
FMT_MOF_CIMC.6.3	FUNC_ACCESSCTRL, FUNC_OCSPPROF
FDP_ACF.1.3 (Iter. 2)	(A none operation has been applied)
FDP_ACF.1.4 (Iter. 2)	(A none operation has been applied)
FDP_ACF_CIMC.2.1	(CIMC personnel private keys are stored in a FIPS 140-2 level 2 Hardware Security Module)
FDP_ACF_CIMC.2.2	See Note about FDP_ACF_CIMC.2.2 requirement in this section
FDP_ACF_CIMC.3.1	See Note about FDP_ACF_CIMC.3.1 requirement in this section
FDP_CIMC_BKP.1.1	FUNC_BACKUP
FDP_CIMC_BKP.1.2	FUNC_BACKUP
FDP_CIMC_BKP.1.3	FUNC_BACKUP
FDP_CIMC_BKP.1.4	FUNC_BACKUP
FDP_CIMC_BKP.2.1	FUNC_I3DSESSION, FUNC_I3DHISTORIC, FUNC_DBIV
FDP_CIMC_BKP.2.2	FUNC_I3DSESSION, FUNC_I3DHISTORIC, FUNC_DBIV
FMT_MTD_CIMC.4.1	See Note about FMT_MTD_CIMC.4.1 requirement in this section
FMT_MTD_CIMC.5.1	See Note about FMT_MTD_CIMC.5.1 requirement in this section
FCS_CKM_CIMC.5.1	(The TOE does not maintain plaintext secret and private keys.)
FDP_CIMC_BKP.1.1	FUNC_BACKUP
FDP_CIMC_BKP.1.2	FUNC_BACKUP
FDP_CIMC_BKP.1.3	FUNC_BACKUP
FDP_CIMC_BKP.1.4	FUNC_BACKUP
FDP_CIMC_BKP.2.1	FUNC_OBFUSCATION, FUNC_I3DSESSION, FUNC_I3DHISTORIC
FDP_CIMC_BKP.2.2	FUNC_OBFUSCATION

Table 6-3. Mapping table between functional requirements and security functions

Note about FDP_ACF_CIMC.2.2 requirement

The only certificate subject private keys that are stored in the TOE are the keys that are been copied by the KeyOne Archive Component (for Key Recovery purposes). These keys are encrypted using Long Term Private Key Protection Key (Component Keys), and this encryption is performed by a cryptographic module, that is a FIPS 140-2 level 2 validated cryptographic module (Administrator of the KeyArchive component).

Note about FDP_ACF_CIMC.3.1 requirement

The only user keys that are stored within the CIMC but not within a FIPS validated cryptographic module, are the subject private keys. These keys are the keys that are been copied by the KeyOne Archive Component (for Key Recovery purposes). These keys are encrypted using a cryptographic module (Administrator of the Key Archive Component) that is a FIPS 140-2 level 2 validated cryptographic module.

Note about FMT_MTD_CIMC.4.1 requirement

The CIMC private keys are stored in a FIPS 140-2 level 3 validated cryptographic module. When these keys are exported to the TOE (shutdown of the system), then they are ciphered by this FIPS 140-2 level 3 Hardware Security Module.

Note about FMT_MTD_CIMC.5.1 requirement

The FMT_MTD_CIMC.5.1 requires that the TSF secret keys stored within the TOE, but not within a FIPS 140-1 validated cryptographic module, be stored in encrypted form. This encryption shall be performed by the FIPS 140-1 validated cryptographic module.

All the secret keys are either stored in FIPS 140-2 level 3 validated cryptographic modules, or they are ciphered using FIPS 140-2 level 3 Hardware Security Module.

Note about FDP_UCT.1.1 requirement

The communication between the Oracle client and the Oracle server, it is protected by means the SSL protocol established between the Oracle client and the Oracle server. This protocol is implemented by the Oracle Database Management System

Functional Requirement	*Security Function*
FUNC_SADG	FAU_GEN.1.1 (Iter. 2), FAU_GEN.1.2 (Iter. 2), FAU_GEN.2.1 (Iter. 2)
FUNC_SELL	FAU_SEL.1.1 (Iter. 2)
FUNC_DBIV	FAU_STG.1.2 (Iter. 2), FDP_CIMC_BKP.2.1
FUNC_CDBC	FAU_STG.4.1 (Iter. 2)
FUNC_I3DSESSION	FPT_STM.1.1 (Iter. 2), FPT_CIMC_TSP.1.1, FPT_CIMC_TSP.1.2, FPT_CIMC_TSP.1.3, FPT_CIMC_TSP.1.4, FDP_SDI_CIMC.3.1,

	FDP_SDI_CIMC.3.2, FDP_CIMC_BKP.2.1
FUNC_I3DHISTORIC	FPT_STM.1.1 (Iter. 2), FPT_CIMC_TSP.1.3, FDP_SDI_CIMC.3.1, FDP_SDI_CIMC.3.2, FDP_CIMC_BKP.2.1
FUNC_ACCESSCTRL	FMT_MOF.1.1 (Iter. 2), FDP_ACC.1.1 (Iter. 2), FDP_ACF.1.1 (Iter. 2), FDP_ACF.1.2 (Iter. 2), FPT_RVM.1.1, FMT_MOF_CIMC.3.2, FMT_MOF_CIMC.3.3, FMT_MOF_CIMC.3.4, FMT_MOF_CIMC.5.2, FMT_MOF_CIMC.5.3, FMT_MOF_CIMC.6.2, FMT_MOF_CIMC.6.3
FUNC_BATCHSIG	FDP_ITT.1.1 (Iter. 3), FDP_SDI_CIMC.3.1, FCO_NRO_CIMC.3.1, FCO_NRO_CIMC.3.2, FPT_ITT.1.1 (Iter. 3)
FUNC_NDCCPSIGCA	FDP_ITT.1.1 (Iter. 3), FPT_ITT.1.1 (Iter. 3), FCO_NRO_CIMC.3.1, FCO_NRO_CIMC.3.2
FUNC_K1SSLTLS	FDP_ITT.1.1 (Iter. 3), FDP_ITT.1.1 (Iter. 4), FDP_SDI_CIMC.3.1, FPT_ITT.1.1 (Iter. 3), FPT_ITT.1.1 (Iter. 4)
FUNC_OBFUSCATION	FPT_ITC.1.1 (Iter. 2), FDP_CIMC_BKP.2.2, FDP_CIMC_BKP.2.1
FUNC_UIDAUT	FIA_UAU.1.1 (Iter. 2), FIA_UAU.1.2 (Iter. 2), FIA_UID.1.1 (Iter. 2) FIA_UAU.1.2 (Iter. 2), FIA_USB.1.1 (Iter. 2)
FUNC_CERTSGENE	FDP_SDI_CIMC.3.1, FDP_CIMC_CER.1.1, FDP_CIMC_CER.1.2, FDP_CIMC_CER.1.4, FMT_MOF_CIMC.3.1
FUNC_PSSINSERT	FDP_SDI_CIMC.3.2
FUNC_PKCS12GENE	FDP_ETC_CIMC.5.1
FUNC_KEYRECOV	FDP_ETC_CIMC.5.1
FUNC_OCSPRES	FDP_CIMC_CSE.1.1, FDP_CIMC_OCSP.1.1, FMT_MOF_CIMC.6.1
FUNC_BATCHVER	FDP_CIMC_CER.1.3, FCO_NRO_CIMC.3.3, FCO_NRO_CIMC.4.1, FCO_NRO_CIMC.4.2
FUNC_CERTREQVER	FDP_CIMC_CER.1.3
FUNC_CRLSGENE	FDP_CIMC_CRL.1.1, FMT_MOF_CIMC.5.1
FUNC_CERTPROF	FMT_MOF_CIMC.3.1, FMT_MOF_CIMC.3.2, FMT_MOF_CIMC.3.3, FMT_MOF_CIMC.3.4
FUNC_REVPROF	FMT_MOF_CIMC.5.1, FMT_MOF_CIMC.5.2, FMT_MOF_CIMC.5.3
FUNC_OCSPPROF	FMT_MOF_CIMC.6.1, FMT_MOF_CIMC.6.2, FMT_MOF_CIMC.6.3
FUNC_BACKUP	FDP_CIMC_BKP.1.1, FDP_CIMC_BKP.1.2, FDP_CIMC_BKP.1.3, FDP_CIMC_BKP.1.4,
FUNC_NDCCPVER	FCO_NRO_CIMC.3.3

Table 6-4. Mapping table between security functions and functional requirements

6.3 Strength Of Functions

This TOE can operate in a range of environments from benign to hostile. The minimum strength of function level for the TOE and IT environment functional security requirements is SOF-basic. Nevertheless the SOF-basic level shall apply except where specific strength of function requirements are needed how it is specified along this section. KeyOne system includes security mechanisms, employed in the authentication service for instance, that use probabilistic or permutational mechanism.

The strength of cryptographic algorithms is outside the scope of the Common Criteria. Strength of function only applies to probabilistic or permutational mechanisms that are non-cryptographic. Consequently, the minimum SOF claim included in this Security Target does not apply to any cryptographic mechanisms with respect to a Common Criteria evaluation.

6.3.1 Authentication Mechanisms

The authentication mechanisms specified in FIA_UAU.1 iterations 1 and 2 shall meet the following strength of function requirements:

1. For each attempt to use the authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur (e.g., guessing a password or PIN, false acceptance error rate of a biometric device, or some combination of authentication methods.)

2. For multiple attempts to use the authentication mechanism during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.

This authentication mechanism is related to the security function FUNC_UIDAUT.

6.3.2 Cryptographic Modules

FIPS 140-2 validated cryptographic modules must perform all cryptographic functions performed by CIMCs. FIPS 140-2 validated cryptographic modules are also required to generate cryptographic keys and to store plaintext private and secret keys.

6.3.2.1 Encryption and FIPS 140-2 Validated Modules

References to FIPS 140-1 refer to the most current version of the standard and the most current version can be found at http://csrc.nist.gov/cryptval.

6.3.2.2 Encryption Algorithms

The encryption specified for:

FAU_STG.1 Protected audit trail storage

FCO_NRO_CIMC.4 Advanced verification of origin

FDP_ACF_CIMC.2 User private key confidentiality protection

FDP_ACF_CIMC.3 User secret key confidentiality protection

FDP_CIMC_BKP.2 Extended CIMC backup and recovery

FDP_ETC_CIMC.4 User private and secret key export

FDP_ETC_CIMC.5 Extended user private and secret key export

FDP_SDI_CIMC.3 Stored public key integrity monitoring and action FMT_MTD_CIMC.4 TSF private key confidentiality protection

FMT_MTD_CIMC.5 TSF secret key confidentiality protection

FMT_MTD_CIMC.6 TSF private and secret key export

FMT_MTD_CIMC.7 Extended TSF private and secret key export

FPT_CIMC_TSP.1 Audit log signing event

FPT_CIMC_TSP.2 Audit log time stamp event

FPT_TST_CIMC.2 Software/firmware integrity test

FPT_TST_CIMC.3 Software/firmware load test shall be performed using a FIPS-approved or recommended algorithm.

6.3.2.3 FIPS 140-2 Validated Cryptographic Modules

Cryptographic modules specified for:

FCS_CKM.1 Cryptographic key generation

FDP_ACF_CIMC.2 User private key confidentiality protection

FDP_ACF_CIMC.3 User secret key confidentiality protection

FDP_ETC_CIMC.4 User private and secret key export

FDP_ETC_CIMC.5 Extended user private and secret key export

FDP_SDI_CIMC.3 Stored public key integrity monitoring and action

FMT_MTD_CIMC.4 TSF private key confidentiality protection

FMT_MTD_CIMC.5 TSF secret key confidentiality protection

FMT_MTD_CIMC.6 TSF private and secret key export

FMT_MTD_CIMC.7 Extended TSF private and secret key export

FPT_CIMC_TSP.1 Audit log signing event shall be validated against FIPS 140-2.

6.3.2.4 Split Knowledge Procedures

Split-knowledge procedures specified in:

FDP_ETC_CIMC.4 User private and secret key export

FDP_ETC_CIMC.5 Extended user private and secret key export

FMT_MTD_CIMC.6 TSF private and secret key export

FMT_MTD_CIMC.7 Extended TSF private and secret key export shall be implemented and validated as specified in FIPS 140-2.

6.3.2.5 Authentication Codes

The authentication code specified in:

FAU_STG.1 Protected audit trail storage

FCO_NRO_CIMC.4 Advanced verification of origin

FDP_CIMC_BKP.2 Extended CIMC backup and recovery

FPT_CIMC_TSP.1 Audit log signing event

FDP_SDI_CIMC.3 Stored public key integrity monitoring and action

FPT_TST_CIMC.2 Software/firmware integrity test

FPT_TST_CIMC.3 Software/firmware load test shall be a FIPS-approved or recommended authentication code.

6.3.2.6 Cryptographic module levels for cryptographic functions that involve private or secret keys

All cryptographic operations performed (including key generation) at the request of the TOE shall be performed in a FIPS 140-1 validated cryptographic module operating in a FIPS-approved or recommended mode of operation.

Table 6-5. FIPS 140-1 Level for Validated Cryptographic Module specifies for each category of use for a private or secret key, the required overall FIPS 140-1 level for the validated cryptographic module. If the CIMC generates certificate subject private keys, the required overall FIPS 140-1 level for *Long Term Private Key Protection* keys shall apply.

Required Overall FIPS 140-1 Level for CIMC Cryptographic Modules

Category of Use	***Security Level 3***
Certificate and Status Signing	
- single party signature	3
- multiparty signature	2

Integrity or Approval Authentication - single approval - dual approval	 2 2
General Authentication	2
Long Term Private Key Protection	3
Long Term Confidentiality	2
Short Term Private key Protection	2
Short Term Confidentiality	1

Table 6-5. FIPS 140-1 Level for Validated Cryptographic Module

6.3.2.7 Cryptographic Functions That Do Not Involve Private or Secret Keys

There are two other cryptographic functions that may be performed in TOE components that do not require private or secret keys. These include:

a) *Hash Generation*: One-way hash functions may be used in the process of signature generation and verification (a signature is typically generated by applying a private key to the hash of the message). The generation of a hash does not require a key. Therefore, hash generation does not have the same confidentiality requirements of other cryptographic functions.

b) *Signature Verification*: Signatures are verified from a message text and a public key. For a cryptographic module that only performs signature verification and/or keyless hash generation functions, the overall required FIPS 140-1 level shall be Level .

6.4 Assurance measures

The security assurance requirements imposed on the TOE are satisfied by the applicable Safelayer Development Methodology, awarded with the ISO 9001:2000 certificate.

This is established by the following documents:

- Company-wide documents that apply to the Software Development Department:
  - QM "Safelayer Quality Manual", Code 28348ACB

    This is the definition of the Company Quality Management System.
  - DMP "Document Management Plan", Code 9D495947

    It establishes the information control rules and procedures for all the documentation handled by the Company.

- Core development process
  - DM "Development Methods", Code A2A7DE72

    This document defines the development loop from a technical point of view.
  - SSL "Software Security Lifecycle", Code 51D94682

    Additional consideration is given to software implementing security measures, and this document complements the basic development process for those cases.
- General disciplines that supports the process
  - QP "Quality Plan", Code D03B789F

    Audits, reviews, and processes and measures to ensure our software development process is excellent and continuously improving.
  - SM "Software Management", Code 3857D336

    Development Management organization, activities and techniques are performed as documented herein.
  - CM "Configuration Management Plan", Code 411A0E26

    Version control, software identification codes, release and branch procedures, all the development history is managed following these procedures.
  - SEM "Security Manual", Code 987EEACF

    Secure products can only be obtained in a secure development environment, and this document regulates the development security procedures.
- Detailed procedure that apply to specific areas and activities
  - GOC "Guía para la Organización de Código Fuente en Safelayer ", Code 2B395E88

    Rules and recommendations to organize source and binary files in a project.

    Compiling flags and rules.
  - CCS "CVS – Control de Versiones", Code 0347E6C0

    How to be a good cvs user.
  - MUATR "Manual de Uso de la ATR", Code E1B0EBCD

    How to use the ATR application.
  - BUG "Safelayer Bugzilla Usage Guidelines", Code A790CDA4

    Describes the usage of the Engineering Change Proposal database and supporting system.

- DSUG "Manual de Usuario del Servidor de Documentación de Safelayer", Code 5C4AC6D9

  User guide of the document management system that gives sense to these links.

- PDP "Product Secure Delivery Procedures", Code 2DB0CC43

  Details the shipping procedures with the appropriate security levels.

Specific TOE documentation is supporting the fulfilment of selected assurance requirements, as identified in the following table:

Assurance Class	Assurance Requirement	Document TiTle
Configuration Management	ACM_AUT.1 ACM_CAP.4 ACM_SCP.2	Configuration Management Manual de uso de la ATR Safelayer Bugzilla Usages Guidelines
Delivery and Operation	ADO_DEL.2	Product Secure Delivery Procedures

	ADO_IGS.1	KeyOne Console 3.0 Administration and User Manual KeyOne CA 3.0 - Administration Manual KeyOne VA 3.0 - Manual Installation and Uninstallation Manual for KeyOne 3.0 Products Signing scripts KeyOne CRL Authority 3.0 Manual KeyOne RA 3.0 Manual KeyOne LRA 3.0 User and Administration Manual KeyOne TSA 3.0 Manual KeyOne 3.0 Logs Registration Administration KeyOne 3.0 Batch Format Description KeyOne 3.0 Master Document KeyOne 3.0 i3D Database Management KeyOne 3.0 Template Textual Specification Syntax
Development	ADV_FSP.2	KeyOne 3.0 Functional Specification
	ADV_HLD.2	KeyOne 3.0 High Level Design

	ADV_LLD.1	KeyOne 3.0 Low Level Design
	ADV_RCR.1	KeyOne 3.0 Functional Specification KeyOne 3.0 High Level Design KeyOne 3.0 Low Level Design
	ADV_SPM.1	KeyOne 3.0 Security Policy
	ADV_IMP.1	KeyOne 3.0 Low Level Design

Guidance Documents	AGD_ADM.1	KeyOne Console 3.0 Administration and User Manual KeyOne CA 3.0 - Administration Manual KeyOne VA 3.0 - Manual Certificate, Keys and CRL Management in KeyOne 3.0 KeyOne CRL Authority 3.0 Manual KeyOne RA 3.0 Manual KeyOne LRA 3.0 User and Administration Manual KeyOne TSA 3.0 Manual KeyOne 3.0 Logs Registration Administration KeyOne 3.0 Batch Format Description KeyOne 3.0 Master Document KeyOne 3.0 i3D Database Management KeyOne 3.0 Template Textual Specification Syntax

	AGD_USR.1	KeyOne Console 3.0 Administration and User Manual KeyOne CA 3.0 – User Manual KeyOne VA 3.0 - Manual Certificate, Keys and CRL Management in KeyOne 3.0 Applications KeyOne CRL Authority 3.0 Manual KeyOne RA 3.0 Manual KeyOne LRA 3.0 User and Administration Manual KeyOne TSA 3.0 Manual KeyOne 3.0 Logs Registration Administration KeyOne 3.0 Batch Format Description KeyOne 3.0 Master Document KeyOne 3.0 i3D Database Management KeyOne 3.0 Template Textual Specification Syntax
Life Cycle Support	ALC_DVS.1	Security Manual Software Security Lifecycle
	ALC_TAT.1	Guía para la organización del código fuente
	ALC_FLR.2	Product Nonconformities Handling Procedures Safelayer Bugzilla Usage Guidelines
	ALC_LCD.1	Software Security Lifecycle
Tests	ATE_COV.2	Quality Assurance-Test Plan
	ATE_FUN.1	Quality Assurance-Test Plan Quality Assurance-Test Description Quality Assurance-Test Result

	ATE_IND.2	Quality Assurance-Test Plan
	ATE_DPT.1	Quality Assurance-Test Plan
Vulnerability Assessment	AVA_SOF.1	KeyOne 3.0 - Security Target KeyOne 3.0 – Strength of TOE Security Function Analysis
	AVA_MSU.2	KeyOne 3.0 Misuse Analysis
	AVA_VLA.2	KeyOne 3.0 Vulnerability Analysis

Table 6-6. Security Assurance Requirements Documentation

6.5 Security functions using probabilistic or permutational mechanisms

The following security functions described in the "*KeyOne 3.0 – Functional Specification*, internal code: 6D6436D9" make use of cryptographic mechanisms.

Security Function
FUNC_DBIV
FUNC_CDBC
FUNC_I3DSESSION
FUNC_I3DHISTORIC
FUNC_BATCHSIG
FUNC_BATCHVER
FUNC_NDCCPSIGCA
FUNC_NDCCPVER
FUNC_K1SSLTLS
FUNC_OBFUSCATION
FUNC_OCSPRES

FUNC_UIDAUT
FUNC_CERTREQVER
FUNC_CERTSGENE
FUNC_PKCSGENE
FUNC_CRLSGENE
FUNC_CERTPROF
FUNC_REVPROF
FUNC_OCSPPROF
FUNC_PSSINSERT
FUNC_KEYRECOV
FUNC_BACKUP

CHAPTER 7

7 Claims

The TOE conforms to the Certificate Issuing and Management Component (CIMC) Protection Profile Security Level 3 (which specifies EAL3 augmented) authored by NIST dated October 31, 2001.

Additionaly KeyOne 3.0 conforms to all the Assurance Requirements for the EAL4 Common Criteria certification level, augmented with ALC_FLR.2. These Assurance Requirements are the following:

- ACM_CAP.4 Generation support and acceptance procedures
- ACM_AUT.1 Partial CM automation
- ALC_LCD.1 Developer defined life-cycle model

The FPT_CIMC_TSP.1.3 security requirement is accomplished because for each modification (addition, update or delete) of a database registry, the i3D mechanism assures the generation of a digital signature that guarantees the database integrity, then KeyOne system works as if it was configured at the maximum frequency, and therefore the frequency most secure (refinement of the FPT_CIMC_TSP.1.3 requirement).

CHAPTER 8

8 Rationale

This section includes the rationale for the functional and assurance requirements specified for the TOE.

The rationale is based on specified objectives, threats, assumptions, and policies.

8.1 Security Objectives Rationale

This section demonstrates that the stated security objectives counter all identified threats, policies, or assumptions.

8.1.1 Security Objectives Coverage

The following tables provide a mapping of security objectives to the environment defined by the threats, policies, and assumptions, illustrating that each security objective covers at least one threat, policy or assumption and that each threat, policy or assumption is covered by at least one security objective. The Table X maps security objectives for the TOE to threats, Table Y maps security objectives for the environment to threats, and Table Z maps security objectives for both the TOE and the environment to threats. Table XX maps the organizational security policies to security objectives. Table YY maps assumptions to IT security objectives, listing which objectives each assumption helps to cover. The items in the tables are ordered alphabetically, sorted on the first column.

IT Security Objective	***Threat***
O.Certificates	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Control unknown source communication traffic	T.Hacker gains access
O.Non-repudiation	T.Sender denies sending information
O.Preservation/trusted recovery of secure state	T.Critical system component fails
O.Sufficient backup storage and effective restoration	T.Critical system component fails, T.User error makes data inaccessible

Table 8-1. Relationships of Security Objectives for the TOE to Threats

Non-IT Security Objective	***Threat***
O.Administrators, Operators, Officers and Auditors guidance documentation	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions , T.Social engineering
O.Competent Administrators, Operators, Officers and Auditors	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.CPS	T.Administrative errors of omission
O.Cryptographic functions	T.Disclosure of private and secret keys, T.Modification of secret/private keys
O.Installation	T.Critical system component fails
O.Lifecycle security	T.Critical system component fails, T.Malicious code exploitation
O.Notify Authorities of Security Issues	T.Hacker gains access
O.Periodically check integrity	T.Malicious code exploitation
O.Physical Protection	T.Hacker physical access
O.Repair identified security flaws	T.Flawed code , T.Critical system component fail
O.Security roles	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Social Engineering Training	T.Social Engineering
O.Trusted path	T.Hacker gains access, T.Message content modification
O.Validation of security function	T.Malicious code exploitation, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions

Table 8-2. Relationship of Security Objectives for the Environment to Threats

Non-IT Security Objective	***Threat***
O.Object and data recovery free from malicious code	T.Modification of secret/private keys, T.Malicious code exploitation
O.Procedures for preventing malicious code	T.Malicious code exploitation, T.Social engineering
O.Protect stored audit records	T.Modification of secret/private keys, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Protect user and TSF data during internal	T.Message content modification,

transfer	T.Disclosure of private and secret keys
O.React to detected attacks	T.Hacker gains access
O.Require inspection for downloads	T.Malicious code exploitation
O.Respond to possible loss of stored audit records	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Restrict actions before authentication	T.Hacker gains access, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Security-relevant configuration management	T.Administrative errors of omission
O.Time stamps	T.Critical system component fails, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Configuration management	T.Critical system component fails, T.Malicious code exploitation
O.Data import/export	T.Message content modification
O.Detect modifications of firmware, software, and backup data	T.User error makes data inaccessible, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Individual accountability and audit records	T.Administrative errors of omission, T.Hacker gains access, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions, T.User abuses authorization to collect and/or send data
O.Integrity protection of user data and software	T.Modification of private/secret keys, T.Malicious code exploitation
O.Limitation of administrative access	T.Disclosure of secret and private keys, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions
O.Maintain user attributes	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions

O.Manage behavior of security functions	T.Critical system component fails, T.Administrators, Operators, Officers and Auditors commit errors or hostile actions

Table 8-3. Relationship of Security Objectives for Both the TOE and the Environment to Threats

Security Policy	***Security Objective***
P.Authorized use of information	O.Auditors review audit logs O.Maintain user attributes O.Restrict actions before authentication O.Security roles O.User authorization management
P.Cryptography	O.Cryptographic functions

Table 8-4. Relationship of Security Policies to Security Objectives

Assumption	***IT Security Objective***
A.Auditors Review Audit Logs	O.Auditors Review Audit Logs
A.Authentication Data Management	O.Authentication Data Management
A.Communications Protection	O.Communications Protection
A.Competent Administrators, Operators, Officersand Auditors	O.Competent Administrators, Operators, Officers and Auditors, O.Installation, O.Security-relevant configuration management, O.User authorization management, O.Configuration Management
A.Cooperative Users	O.Cooperative Users
A.CPS	O.CPS, O.Security-relevant configuration management, O.User authorization management, O.Configuration Management
A.Disposal of Authentication Data	O.Disposal of Authentication Data
A.Malicious Code Not Signed	O.Procedures for preventing malicious code,

	O.Require inspection for downloads, O.Malicious Code Not Signed
A.Notify Authorities of Security Issues	O.Notify Authorities of Security Issues
A.Operating System	O.Operating System
A.Physical Protection	O.Physical Protection
A.Social Engineering Training	O.Social Engineering Training
A.NTP Client	O.Time Stamp

Table 8-5. Relationship of Assumptions to IT Security Objectives

8.1.2 Security Objectives Sufficiency

The following discussions provide information regarding:

1. Why the identified security objectives provide for effective countermeasures to the threats;

2. Why the identified security objectives provide complete coverage of each organizational security policy;

3. Why the identified security objectives uphold each assumption.

8.1.2.1 Threats and Objectives Sufficiency

8.1.2.1.1 Authorized users

T.Administrative errors of omission addresses errors that directly compromise organizational security objectives or change the technical security policy enforced by the system or application.

It is countered by:

> **O.CPS** provides Administrators, Operators, Officers, and Auditors with information regarding the policies and practices used by the system. Providing this information ensures that these authorized users of the system are aware of their responsibilities, thus reducing the likelihood that they will fail to perform a security-critical operation.
>
> **O.Individual accountability and audit records** provides individual accountability for audited events. Each user is uniquely identified so that auditable actions can be traced to a user. Audit records provide information about past user behavior to an authorized individual through system mechanisms. These audit records will expose administrators that fail to perform security-critical operations so they can be held accountable.
>
> **O.Security-relevant configuration management** ensures that system security policy data and enforcement functions, and other security-relevant configuration data are managed and updated. This ensures that they are

consistent with organizational security policies and that all changes are properly tracked and implemented.

T.User abuses authorization to collect and/or send data addresses the situation where an authorized user abuses granted authorizations by browsing files in order to collect data and/or violates export control policy by sending data to a recipient who is not authorized to receive the data.

It is countered by:

O.Individual accountability and audit records provides individual accountability for audited events. Each user is uniquely identified so that auditable actions can be traced to a user. Audit records provide information about past user behavior to an authorized individual through system mechanisms. This audit records will expose users who abuse their authorized to collect and/or send data.

T.User error makes data inaccessible addresses a user accidentally deleting user data. Consequently, the user data is inaccessible. Examples include the following:

– User accidentally deletes data by striking the wrong key on the keyboard or by striking the enter key as an automatic response.

– User does not understand the implications of the prompt at hand and inadvertently gives a response that deletes user data.

– User misunderstands a system command and issues a command that unintentionally deletes user data.

It is countered by:

O.Sufficient backup storage and effective restoration ensures that there is sufficient backup storage and effective restoration to recreate the system, when required. This ensures that user data is available from backup, even if the current copy is accidentally deleted.

O.Detect modifications of firmware, software, and backup data ensures that if the backup components have been modified, that it is detected. If modifications of backup data can not be detected, the backup copy is not a reliable source for restoration of user data.

T.Administrators, Operators, Officers and Auditors commit errors or hostile actions addresses:

- Errors committed by administrative personnel that directly compromise organizational security objectives, change the technical security policy enforced by the system or application, or

- Malicious obstruction by administrative personnel of organizational security objectives or modification of the system's configuration to allow security violations to occur.

It is countered by:

O.Competent Administrators, Operators, Officers and Auditors ensures that users are capable of maintaining effective security practices. This reduces the likelihood that they will commit errors.

O.Administrators, Operators, Officers and Auditors guidance documentation which deters administrative personnel errors by providing adequate guidance.

O.Certificates ensures that certificates, certificate revocation lists, and certificate status information are valid. The validation of information provided by Officers that is to be included in certificates helps to prevent improperly entered information from appearing in certificates.

O.Detect modifications of firmware, software, and backup data ensures that if the backup components have been modified, that it is detected.

O.Individual accountability and audit records provides individual accountability for audited events. Each user is uniquely identified so that auditable actions can be traced to a user. Audit records provide information about past user behavior to an authorized individual through system mechanisms. These audit records will expose administrators that perform inappropriate operations so they can be held accountable.

O.Limitation of administrative access. The administrative functions are designed in such a way that administrative personnel do not automatically have access to user objects, except for necessary exceptions. In general, the exceptions tend to be role specific. Limiting the set of operations that a user may perform limits the damage that a user may cause.

O.Maintain user attributes. Maintains a set of security attributes (which may include group membership, access rights, etc.) associated with individual users in addition to user identity. This prevents users from performing operations that they are not authorized to perform.

O.Manage behavior of security functions provides management controls/functions for security mechanisms. This ensures that security mechanisms which protect against hostile users are properly configured.

O.Protect stored audit records ensures that audit records are protected against unauthorized access, modification, or deletion to provide for traceability of user actions.

O.Respond to possible loss of stored audit records ensures that only auditable events executed by the Auditor shall be audited if the audit trail is full. This ensures that operations that are performed by users other than the Auditor are audited and so can be detected.

O.Restrict actions before authentication ensures that only a limited set of actions may be performed before a user is authenticated.

O.Security roles ensures that security-relevant roles are specified and that users are assigned to one (or more) of the defined roles. This prevents users from performing operations that they are not authorized to perform.

O.Time stamps ensures that time stamps are provided to verify a sequence of events. This allows the reconstruction of a timeline of events when performing an audit review.

O.Validation of security function. Ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures such as underlying machine testing and integrity checks.

8.1.2.1.2 System

T.Critical system component fails addresses the failure of one or more system components that results in the loss of system-critical functionality. This threat is relevant when there are components that may fail due to hardware and/or software imperfections and the availability of system functionality is important.

It is countered by:

O.Configuration management assures that a configuration management program is implemented. The configuration management program includes configuration identification and change control. This ensures that critical system components do not fail as a result of improper configuration.

O.Installation ensures that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security. This ensures that critical system components do not fail as a result of improper installation.

O.Manage behavior of security functions provides management controls/functions for security mechanisms. This ensures that critical system components do not fail as a result of improper configuration of security mechanisms.

O.Preservation/trusted recovery of secure state ensures that the system remains in a secure state throughout operation in the presence of failures and subsequent system recovery. This objective is relevant when system failures could result in insecure states that, when the system returns to operational mode (or continues to operate), could lead to security compromises.

O.Sufficient backup storage and effective restoration ensures that there is sufficient backup storage and effective restoration to recreate the system, when required. This ensures that data is available from backup, even if the current copy is lost through failure of a system component (e.g., a disk drive).

O.Time stamps provides time stamps to ensure that the sequencing of events can be verified. If the system must be reconstructed, it may be necessary to establish the order in which transactions were performed to return the system to a state consistent with the state when a critical component failed..

O.Lifecycle security provides tools and techniques that are used throughout the development phase reducing the likelihood of hardware or software imperfections. **O.Lifecycle security** also addresses the detection and resolution of flaws discovered during the operational phase that may result in failure of a critical system component.

O.repair identified security flaws. The vendor repairs security flaws that have been identified by a user. Such security flaws may result in critical system component failures if not repaired.

T.Flawed code addresses accidental or deliberate flaws in code made by the developer. Examples of accidental flaws are lack of engineering detail or bad design. An example of a deliberate flaw would be the inclusion of a trapdoor for later entry into the TOE.

It is countered by:

> **O.Repair identified security flaws** ensures that identified security flaws are repaired.

T.Malicious code exploitation addresses the threat where an authorized user, IT system, or hacker downloads and executes malicious code, which causes abnormal processes that violate the integrity, availability, or confidentiality of the system assets. The execution of malicious code is done through a triggering event.

It is countered by:

> **O.Configuration management** assures that a configuration management program is implemented. The configuration management program includes configuration identification and change control. This ensures that malicious code is not introduced during the configuration process.
>
> **O.Integrity protection of user data and software** ensures that appropriate integrity protection is provided for user data and software. This prevents malicious code from attaching itself to user data or software.
>
> **O.Object and data recovery free from malicious code** ensures that the system recovers to a viable state after malicious code has been introduced and damage has occurred. The malicious code, e.g., virus or worm, is removed as part of the process.
>
> **O.Periodically check integrity** ensures that periodic integrity checks are performed on both system and software. If these checks fail, malicious code may have been introduced into the system.
>
> **O.Procedures for preventing malicious code** provides a set of procedures and mechanisms that work to prevent incorporation of malicious code into the system.
>
> **O.Require inspection for downloads** ensures that software that is downloaded/transferred is inspected prior to being made operational.
>
> **O.Validation of security function**. Ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures such as underlying machine testing and integrity checks.
>
> **O.Lifecycle security** provides tools and techniques that are used throughout the development phase, reducing the likelihood that malicious code was included in the product by the developer.
>
> **O.Lifecycle security** also addresses the detection and resolution of flaws discovered during the operational phase, such as modifications of components by malicious code.

T.Message content modification addresses the situation where a hacker modifies information that is intercepted from a communications link between two unsuspecting entities before passing it on to the intended recipient. Several kinds of

modification are possible: modification of a single message, deletion or reordering of selected messages, insertion of bogus messages, replay of previous messages, and modification of accompanying message security attributes.

It is countered by:

O.Data Import/Export protects data when being transmitted to or from the TOE. Protection of data in transit permits the TOE or the external user to detect modified messages, message replay, or fraudulent messages.

O.Protect user and TSF data during internal transfer protects data being transmitted between separated parts of the TOE. Protection of data in transit permits the TOE to detect modified messages, message replay, or fraudulent messages.

O.Trusted path ensures that a trusted path is established between the user and the system. The trusted path protects messages from interception or modification by a hacker.

8.1.2.1.3 Cryptography

T.Disclosure of private and secret keys addresses the unauthorized disclosure of secret and/or private keys.

It is countered by:

O.Administrators, Operators, Officers and Auditors guidance documentation ensures that adequate documentation on securely configuring and operating the CIMC is available to Administrators, Operators, Officers and Auditors. This documentation will minimize errors committed by those users.

O.Cryptographic functions ensures that TOE implements approved cryptographic algorithms for encryption/decryption, authentication, and signature generation/verification; approved key generation techniques and uses validated cryptographic modules. Use of validated cryptographic modules ensures that cryptographic keys are adequately protected when they are stored within cryptographic modules.

O.Limitation of administrative access. The administrative functions are designed in such a way that administrative personnel do not automatically have access to user objects, except for necessary exceptions. In general, the exceptions tend to be role specific. Limiting the number of users who have access to cryptographic keys reducing the likelihood of unauthorized disclosure.

O.Protect user and TSF data during internal transfer protects private and secret keys from unauthorized disclosure during transmission between separated parts of the TOE.

T.Modification of private/secret keys addresses the unauthorized revision of a secret and/or private key.

It is countered by:

O.Cryptographic functions ensures that TOE implements approved cryptographic algorithms for encryption/decryption, authentication, and

signature generation/verification; approved key generation techniques and uses validated cryptographic modules. Use of validated cryptographic modules ensures that cryptographic keys are adequately protected when they are stored within cryptographic modules.

O.Integrity protection of user data and software that ensures that appropriate integrity protection is provided for secret and private keys.

O.Object and data recovery free from malicious code ensures that the system recovers to a viable state after malicious code has been introduced and damage has occurred. If the malicious code cause private or secret keys to be revised in an unauthorized manner, this objective ensures that they are recovered to their correct values.

O.Protect stored audit records ensures that audit records are protected against unauthorized access, modification, or deletion to provide for traceability of user actions. This objective ensures that modifications to private and secret keys can be detected through the audit trail.

T.Sender denies sending information addresses the situation where the sender of a message denies sending the message to avoid accountability for sending the message and for subsequent action or inaction.

It is countered by:

O.Non-repudiation which ensures that the sender/originator of a message cannot successfully deny sending the message to the recipient.

8.1.2.1.4 External Attacks

T.Hacker gains access addresses:

- Weak system access control mechanisms or user attributes

- Weak implementation methods of the system access control

- Vulnerabilities found in system or application code that allow a hacker to break into a system undetected.

It is countered by:

O.Restrict actions before authentication ensures that only a limited set of actions may be performed before a user is authenticated. This prevents a hacker who is unable to circumvent the access control mechanisms from performing security-relevant operations.

O.Control unknown source communication traffic ensures that communication traffic from an unknown source is controlled (e.g., rerouted or discarded) to prevent potential damage. Various kinds of hacker attacks can be detected or prevented by rerouting or discarding suspected hacker traffic.

O.Individual accountability and audit records provides individual accountability for audited events. Each user is uniquely identified so that auditable actions can be traced to a user. Audit records provide information about past user behavior to an authorized individual through system

mechanisms. This allows for the detection of unauthorized activity. Once detected, the damage resulting from such activity can be eliminated or mitigated.

O.Notify Authorities of Security Issues ensures that proper authorities are notified regarding any security issues that impact their systems. This minimizes the potential for the loss or compromise of data.

At this Security Level it is also countered by:

O.React to detected attacks ensures that automated notification or other reactions to the TSFdiscovered attacks is implemented in an effort to identify attacks and to create an attack deterrent. This objective is relevant if actions that the organization deems essential also pose a potential attack that could be exploited.

At this Security Level it is also countered by:

O.Trusted path ensures that a trusted path is established between the user and the system. The trusted path is used to protect authentication data, thus reducing the likelihood that a hacker can masquerade as an authorized user.

T.Hacker physical access addresses the threat where an individual exploits physical security weaknesses to gain physical control of system components.

It is countered by:

O.Physical Protection ensures that physical access controls are sufficient to thwart a physical attack on system components.

T.Social Engineering addresses the situation where a hacker uses social engineering techniques to gain information about system entry, system use, system design, or system operation.

It is countered by:

O.Administrators, Operators, Officers and Auditors guidance documentation which deters administrative personnel errors by providing adequate guidance.

O.Procedures for preventing malicious code provides a set of procedures and mechanisms that work to prevent incorporation of malicious code into the system. The introduction of malicious code into the system may be a goal of the social engineering attack.

O.Social Engineering Training which ensures that general users, Administrators, Operators, Officers, and Auditors are trained in techniques to thwart social engineering attacks.

8.1.2.2 *Policies and Objectives Sufficiency*

P.Authorized use of information establishes that information is used only for its authorized purpose(s).

This is addressed by the following objectives: **O.Maintain user attributes**, **O.Restrict actions before authentication**, **O.Security roles**, and **O.User authorization management**. **O.Restrict actions before authentication** ensures that the capability to perform security-relevant operations is limited to those who have been authorized to perform those operations. **O.Maintain user attributes**, **O.Security roles**, and **O.User authorization management** ensure that users are only authorized to perform those operations that are necessary to perform their jobs. Finally, **O.Auditors review audit logs** deters users from misusing the authorizations they have been provided.

P.Cryptography establishes that accepted cryptographic standards and operations shall be used in the design of the TOE. This is addressed by **O.Cryptographic functions** which ensures that such standards are used.

8.1.2.3 *Assumptions and Objectives Sufficiency*

8.1.2.3.1 Personnel

A.Auditors Review Audit Logs establishes that audit logs are necessary for security-relevant events and that they must be reviewed by auditors. This is addressed by **O.Auditors Review Audit Logs**, which ensures that security-relevant events recorded in audit logs are reviewed by auditors.

A.Authentication Data Management establishes that management of user authentication data is external to the TOE. This is addressed by **O.Authentication Data Management**, which ensures that users modify their authentication data in accordance with appropriate security policy.

A.Competent Administrators, Operators, Officers and Auditors establishes that security of the TOE is dependent upon those that manage it. This is addressed by the following objectives:

- **O.Competent Administrators, Operators, Officers and Auditors**, which ensures that the system managers will be competent in its administration.
- **O.Installation**, which ensures that the responsible for the TOE ensures that the TOE is delivered, installed, managed and operated in a manner which maintains IT security.
- **O.Security-relevant configuration management**, which ensures that the organizational security policies are consistent with the system security policy data, enforcement functions, and other security-relevant configuration data.
- **O.Configuration Management**, which ensures that the system connectivity (software, hardware and firmware) and components (software, hardware and firmware) are identified, that the configuration data are audited, and that the changes to the configuration items are controlled.

A.CPS establishes that Administrators, Operators, Officers, and Auditors are familiar with the CP and CPS under which the TOE is operated. This is addressed by the following objectives:

- **O.CPS,** which ensures that Administrators, Operators, Officers, and Auditors are familiar with the CP and CPS under which the TOE is operated.

- **O.Security-relevant configuration management**, which ensures that the organizational security policies are consistent with the system security policy data, enforcement functions, and other security-relevant configuration data.

- **O.User authorisation management**, which ensures that the user authorisation and privilege data are consistent with organizational security and personnel policies.

- **O.Configuration Management**, which ensures that the system connectivity (software, hardware and firmware) and components (software, hardware and firmware) are identified, that the configuration data are audited, and that the changes to the configuration items are controlled. **A.Malicious Code Not Signed** establishes that code not designed for the TOE will not be signed by a trusted party. This is addressed by the following objectives:

- **O.Malicious Code Not Signed,** which ensures that code must be signed by a trusted party or it will not be loaded onto the system.

- **O.Procedures for preventing malicious code,** which incorporates malicious code prevention procedures and mechanisms.

- **O.Require inspection for downloads,** which ensures inspection of downloads/transfers.

A.Disposal of Authentication Data establishes that users shall not retain access to the system after their authorization has been removed. This is addressed by **O.Disposal of Authentication Data**, which ensures that access to the system will be denied after a user's privileges have been removed.**A.Malicious Code Not Signed** establishes that code not designed for the TOE will not be signed by a trusted party. This is addressed by **O.Malicious Code Not Signed,** which ensures that code must be signed by a trusted party or it will not be loaded onto the system.

A.Notify Authorities of Security Issues establishes that users notify proper authorities of any security issues that impact their systems to minimize the potential for the loss of compromise of data. This is addressed by **O.Notify Authorities of Security Issues** which ensures that user notify proper authorities of any security issues that impact their systems.

A.Social Engineering Training establishes that individuals will attempt to gain access to the system using social engineering practices. This is addressed by **O.Social Engineering Training**, which ensures that all users will be training to thwart social engineering attacks.

A.Cooperative Users establishes that a secure IT environment is required to securely operate the TOE, and that users must work within the constraints of that environment. This is addressed by **O.Cooperative Users**, which ensures that users will cooperate with the constraints established.

8.1.2.3.2 Connectivity

A.Operating System establishes that an insecure operating system will compromise system security. This is addressed by **O.Operating System**, which ensures that an operating system that meets security requirements recommended by the National Institute of Standards and Technology will be used.

A.NTP Client establishes that an erroneous time provided by the system clock where the components of the TOE are running, will compromise system security. This is addressed by **O.Time Stamp**, which provides time stamps.

8.1.2.3.3 Physical

A.Communications Protection establishes that the communications infrastructure is outside the TOE. This is addressed by **O.Communications Protection**, which ensures that adequate physical protections are afforded the necessary communications infrastructure.

A.Physical Protection establishes that physical modification of the TOE hardware, software, and firmware will compromise system security. This is addressed by **O.Physical Protection**, which ensures that adequate physical protection will be provided.

8.2 Security Requirements Rationale

This section provides the rationale for necessity and sufficiency of security requirements, demonstrating that each of the security objectives is addressed by at least one security requirement, and that every security requirement is directed toward solving at least one objective.

8.2.1 Security Requirements Coverage

The following tables provide a mapping of the relationships of security requirements to objectives, illustrating that each security requirement covers at least one objective and that each objective is covered by at least one security requirement. The first table in this section, Table 8.6, addresses the mapping of security functional requirements to security objectives. The second table, Table 8.7, addresses the mapping of security assurance requirements to security objectives.

Functional Requirement	***Objective***
FAU_GEN.1 Audit data generation (iterations 1and 2)	O.Individual accountability and audit records
FAU_GEN.2 User identity association (iterations 1 and 2)	O.Individual accountability and audit records
FAU_SAR.1 Audit review	O.Individual accountability and audit records
FAU_SAR.3 Selectable audit review	O.Individual accountability and audit records
FAU_SEL.1 Selective Audit (iterations 1 and 2)	O.Individual accountability and audit records
FAU_STG.1 Protected audit trail storage (iterations 1 and 2)	O.Protect stored audit records
FAU_STG.4 Prevention of audit data loss (iterations 1 and 2)	O.Respond to possible loss of stored audit records
FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin	O.Non-repudiation, O.Control unknown source communication traffic

FCO_NRO_CIMC.4 Advanced verification of origin	O.Non-repudiation
FCS_CKM.1 Cryptographic key generation	O.Cryptographic functions
FCS_CKM.4 Cryptographic key	O.Procedures for preventing malicious code, O.React to detected attacks
FCS_CKM_CIMC.5 CIMC private and secret key zeroization	O.Procedures for preventing malicious code, O.React to detected attacks
FCS_COP.1 Cryptographic operation	O.Cryptographic functions
FDP_ACC.1 Subset access control (iterations 1 and 2)	O.Limitation of administrative access
FDP_ACF.1 Security attribute based access control (iterations 1 and 2)	O.Limitation of administrative access
FDP_ACF_CIMC.2 User private key confidentiality protection	O.Certificates, O.Procedures for preventing malicious code
FDP_ACF_CIMC.3 User secret key confidentiality protection	O.Certificates, O.Procedures for preventing malicious code
FDP_CIMC_BKP.1 CIMC backup and recovery	O.Object and data recovery free from malicious code, O.Preservation/trusted recovery of secure state, O.Sufficient backup storage and effective restoration
FDP_CIMC_BKP.2 Extended CIMC backup and recovery	O.Detect modifications of firmware, software, and backup data, O.Object and data recovery free from malicious code
FDP_CIMC_CER.1 Certificate Generation	O.Certificates
FDP_CIMC_CRL.1 Certificate revocation list validation	O.Certificates
FDP_CIMC_CSE.1 Certificate status export	O.Certificates
FDP_CIMC_OCSP.1 OCSP basic response validation	O.Certificates
FDP_ETC_CIMC.5 Extended user private and gsecret key export	O.Data import/export
FDP_ITT.1 Basic internal transfer protection (iterations 1 and 3)	O.Integrity protection of user data and software, O.Protect user and TSF data during internal transfer
FDP_ITT.1 Basic internal transfer protection (iterations 2 and 4)	O.Protect user and TSF data during internal transfer
FDP_SDI_CIMC.3 Stored public key integrity monitoring and action	O.Integrity protection of user data and software
FDP_UCT.1 Basic data exchange confidentiality (iterations 1 and 2)	O.Data import/export
FIA_AFL.1 Authentication failure handling	O.React to detected attacks
FIA_ATD.1 User attribute definition	O.Maintain user attributes
FIA_UAU.1 Timing of authentication (iterations 1	O.Limitation of administrative access,

and 2)	O.Restrict actions before authentication
FIA_UID.1 Timing of identification (iterations 1and 2)	O.Individual accountability and audit records, O.Limitation of administrative access
FIA_USB.1 User-subject binding (iterations 1 and 2)	O.Maintain user attributes
FMT_MOF.1 Management of security functions behavior (iterations 1 and 2)	O.Configuration management, O.Manage behavior of security functions, O.Security-relevant configuration management
FMT_MOF_CIMC.3 Extended certificate profile management	O.Configuration management
FMT_MOF_CIMC.5 Extended certificate revocation list profile management	O.Configuration management
FMT_MOF_CIMC.6 OCSP Profile Management	O.Configuration management
FMT_MSA.1 Management of security attributes	O.Maintain user attributes, O.User authorization management
FMT_MSA.2 Secure security attributes	O.Security-relevant configuration management
FMT_MSA.3 Static attribute initialisation	O.Security-relevant configuration management
FMT_MTD.1 Management of TSF data	O.Individual accountability and audit records, O.Protect stored audit records
FMT_MTD_CIMC.4 TSF private key confidentiality protection	O.Detect modifications of firmware, software, and backup data, O.Integrity protection of user data and software
FMT_MTD_CIMC.5 TSF secret key confidentiality protection	O.Detect modifications of firmware, software, and backup data, O.Integrity protection of user data and software
FMT_MTD_CIMC.7 Extended TSF private and secret key export	O.Data import/export
FMT_SMR.2 Restrictions on security roles	O.Security roles
FMT_SMF.1 Specification of Management Functions	O.Security roles, O.Maintain user attributes, O.Manage behaviour of security functions
FPT_AMT.1 Abstract machine testing	O.Periodically check integrity, O.Validation of security function
FPT_CIMC_TSP.1 Audit log signing event	O.Protect stored audit records
FPT_ITC.1 Inter-TSF confidentiality during transmission (iterations 1 and 2)	O.Data import/export
FPT_ITT.1 Basic internal TSF data transfer protection (iterations 1-4)	O.Protect user and TSF data during internal transfer
FPT_RVM.1 Non-bypassability of the TSP (iteration 1)	O.Operating System

FPT_RVM.1 Non-bypassability of the TSP (iteration 2)	O.Limitation of administrative access
FPT_SEP.1 TSF domain separation	O.Operating System
FPT_STM.1 Reliable time stamps (iterations 1 and 2)	O.Individual accountability and audit records, O.Time stamps
FPT_TST_CIMC.2 Software/firmware integrity test	O.Detect modifications of firmware, software, and backup data, O.Integrity protection of user data and ¡software, O.Object and data recovery free from malicious code, O.Periodically check integrity, O.Procedures for preventing malicious code, O.Validation of security function
FPT_TST_CIMC.3 Software/firmware load test	O.Integrity protection of user data and software, O.Object and data recovery free from malicious code, O.Periodically check integrity, O.Require inspection for downloads
FTP_TRP.1 Trusted path	O.Trusted path
FPT_ACC.1 Access Control	O.Protect stored audit records

Table 8-6. Security Functional Requirements Related to Security Objectives

Assurance Requirement	***Objective***
ACM_AUT.1 Automation	EAL4, O.Configuration management
ACM_CAP.4 Generation support andacceptance procedures	EAL 4, O.Configuration management
ACM_SCP.2 Problem tracking CM Coverage	EAL 4, O.Configuration management
ADO_DEL.2 Detection of modification	EAL 4
ADO_IGS.1 Installation, Generation, and Start-up Procedures	EAL 4, O.Installation
ADV_FSP.2 Fully defined external interfaces	EAL 4, O.Lifecycle security
ADV_HLD.2 Security enforcing high-level design	EAL 4, O.Lifecycle security
ADV_IMP.1 Subset of the implementation of the TSF	EAL 4, O.Lifecycle security
ADV_LLD.1 Descriptive low-level design	EAL 4, O.Lifecycle security

ADV_RCR.1 Informal Correspondence Demonstration	O.Lifecycle security, EAL 4
ADV_SPM.1 Informal TOE security policy model	EAL 4, O.Lifecycle security
AGD_ADM.1 Administrator Guidance	O.Administrators, Operators, Officers and Auditors guidance documentation, O.Auditors Review Audit Logs, O.Competent Administrators, Operators, Officers and Auditors, O.Configuration Management, O.Installation, O.Malicious Code Not Signed, O.Procedures for preventing malicious code, O.Require inspection for downloads, O.Security-relevant configuration management, O.User authorization management, EAL 4
AGD_USR.1 User Guidance	O.Administrators, Operators, Officers and Auditors guidance documentation, O.Malicious Code Not Signed, O.Procedures for preventing malicious code, O.Require inspection for downloads, EAL 4
ALC_DVS.1 Identification of security measures	EAL 4
ALC_FLR.2 Flaw reporting procedures	O.Lifecycle security O.Repair identified security flaws, EAL4
ALC_LCD.1 Developer defined life-cycle model	EAL 4
ALC_TAT.1 Well-defined development tools	EAL 4
ATE_COV.2 Analysis of coverage	EAL 4
ATE_DPT.1 Testing - High-Level Design	EAL4
ATE_FUN.1 Functional testing	EAL 4

ATE_IND.2 Independent Testing	EAL 4
AVA_MSU.2 Validation of analysis	EAL 4
AVA_SOF.1 Strength of TOE Security Function Evaluation	EAL 4
AVA_VLA.2 Independent vulnerability analysis	EAL 4

Table 8-7. Security Assurance Requirements Related to Security Objectives

8.2.2 Security Requirements Sufficiency

Security Objectives for the TOE

8.2.2.1 Authorized Users

O.Certificates is provided by **FDP_CIMC_CER.1 (Certificate Generation)** which ensures that certificates are valid, and **FDP_CIMC_CRL.1 (Certificate revocation list validation**), **FDP_CIMC_CSE.1 (Certificate status export)**, and **FDP_CIMC_OCSP.1 (OCSP basic response validation**) which ensure that certificate revocation lists and certificate status information are valid. In the case that the TOE maintains a copy of the certificate subject's private key, **FDP_ACF_CIMC.2 (User private key confidentiality protection)** ensures that the certificate is not invalidated by the disclosure of the private key by the TOE. In the case that a secret key is used by the certificate subject as an authenticator in requesting a certificate, **FDP_ACF_CIMC.3 (User secret key confidentiality protection)** ensures that an attacker can not obtain a bad certificate by obtaining a user's authenticator from the TOE and then using that authenticator to obtain a bad certificate.

8.2.2.2 System

O.Preservation/trusted recovery of secure state is provided by **FDP_CIMC_BKP.1 (CIMC backup and recovery)** which cover the requirement that the state of the system be preserved so that it can be recovered in the event of a secure component failure.

O.Sufficient backup storage and effective restoration is provided by **FDP_CIMC_BKP.1 (CIMC backup and recovery)** which cover the requirement that sufficient backup data is created and stored and that an effective restoration procedure is provided.

8.2.2.3 External Attacks

O.Control unknown source communication traffic is provided by **FCO_NRO_CIMC.3 (Enforced proof of origin and verification of origin)** which covers the requirement that the TOE discard messages from an unknown source that contain security-relevant information.

8.2.2.4 Cryptography

O.Non-repudiation is provided by **FCO_NRO_CIMC.3 (Enforced proof of origin and verification of origin)** which covers the requirement that messages containing security-

relevant data are not accepted by the TOE unless they contain evidence of origin and **FCO_NRO_CIMC.4 (Advanced verification of origin)** which covers the requirement that digital signatures be used so that the evidence of origin for a message may be verified by a third-party.

Non-IT Security Objectives for the Environment

O.Administrators, Operators, Officers and Auditors guidance documentation is provided by **AGD_ADM.1 (Administrator Guidance)** and **AGD_USR.1 (User Guidance)** which ensure that adequate guidance on the secure operation of the TOE is provided to Administrators, Operators, Officers, and Auditors.

O.Auditors Review Audit Logs is provided by **A.Auditors Review Audit Logs** which ensures that auditors review the audit logs. It is also supported by **AGD_ADM.1 (Administrator Guidance)** which ensures that Auditors are provided with the information they need to understand the contents of the audit logs.

O.Authentication Data Management is provided by **A.Authentication Data Management** which covers the requirement that an authentication data management policy be enforced.

O.Communications Protection is provided by **A.Communications Protection** which covers the requirement that the system be adequately physically protected against loss of communications.

O.Competent Administrators, Operators, Officers and Auditors is provided **by A.Competent Administrators, Operators, Officers and Auditors** which covers the requirement that Administrators, Operators, Officers, and Auditors be capable of managing the TOE and the security of the information it contains. It is also supported by **AGD_ADM.1 (Administrator Guidance)** which ensures that Administrators, Operators, Officers, and Auditors are provided with the information they need to properly manage the TOE and its security functionality.

O.CPS is provided by **A.CPS** which covers the requirement that Administrators, Operators, Officers, and Auditors be familiar with the CP and CPS under which the TOE is operated.

O.Installation is provided by **ADO_IGS.1 (Installation, Generation, and Start-up Procedures)** and **AGD_ADM.1 (Administrator Guidance)** which cover the requirement that Administrators, Operators, Officers, and Auditors be provided with documentation describing the procedures necessary to securely install and operate the TOE. **A.Competent Administrators, Operators, Officers and Auditors** covers the requirement that competent Administrators, Operators, Officers, and Auditors, who are capable of securely managing the TOE, are used.

O.Malicious Code Not Signed is provided by **A.Malicious Code Not Signed** which covers the requirement that malicious code destined for the TOE is not signed by a trusted entity. It is also supported by **AGD_ADM.1 (Administrator Guidance)** and **AGD_USR.1 (User Guidance)** which ensure that entities that are trusted to sign code are aware of their responsibilities.

O.Notify Authorities of Security Issues is provided by **A.Notify Authorities of Security Issues** which covers the requirement that proper authorities be notified of any security issues that impact their systems.

O.Physical Protection is provided by **A.Physical Protection** which covers the requirement that TOE hardware, software, and firmware critical to security policy enforcement be protected from unauthorized physical modification.

O.Social Engineering Training is provided by **A.Social Engineering Training** which covers the requirement that general users, administrators, operators, officers, and auditors are trained in techniques to thwart social engineering attacks.

O.Cooperative Users is provided by **A.Cooperative Users** which covers the requirement that users act in a cooperative manner.

O.Lifecycle security is provided by **ADV_FSP.2 (Fully defined external interfaces)**, **ADV_HLD.2 (Security enforcing high-level design)**, **ADV_LLD.1 (Descriptive low-level design)**, **ADV_RCR.1 (Informal correspondence demonstration)**, and **ADV_SPM.1 (Information TOE security policy model)** which cover the requirement that security is designed into the CIMC. **ALC_FLR.2 (Flaw reporting procedures)** that flaws are detected and resolved during the operational phase.

O.Repair identified security flaws is provided by **ALC_FLR.2 (Flaw reporting procedures)** which cover the requirement that vendor repair security flaws that have been identified by a user.

O.Disposal of Authentication Data is provided by **A.Disposal of Authentication Data,** which covers the requirement that authentication data be disposed of properly after access has been removed.

IT Security Objectives for the Environment

O.Cryptographic functions is provided by **FCS_CKM.1 (Cryptographic key generation)** and **FCS_COP.1 (Cryptographic operation)** which cover the requirement that approved algorithms be used for encryption/decryption, authentication, and signature generation/verification and that approved key generation techniques be used.

O.Operating System is provided by **A.Operating System** which covers the requirement that the operating system(s) on which the TSF operates provides security functions required by the CIMC to counter the perceived threats for the appropriate Security Level. It is also supported by **FPT_RVM.1 (Non-bypassability of the TSP) (iteration 1)** and **FPT_SEP.1 (TSF domain separation)** which ensure that the operating system(s) on which the TSF operates provides domain separation and non-bypassability.

O.Periodically check integrity is provided by **FPT_AMT.1 (Abstract machine testing)** which covers the requirement provide periodic integrity checks on the system and **FPT_TST_CIMC.2 (Software/firmware integrity test)** and **FPT_TST_CIMC.3 (Software/firmware load test)** cover the requirement to periodically check the integrity of software.

O.Security roles is provided by **FMT_SMR.2 (Restrictions on security roles)** which covers the requirement that a set of security roles be maintained and that users be associated with those roles, and **FMT_SMF.1 (Specification of Management Functions)** which covers the requirement that and specific management functions are provided, like the management of users and permissions of access on the part of the users, and the administration of useres authentication.

O.Validation of security function is provided by **FPT_AMT.1 (Abstract machine testing)** which covers the requirement to ensure that security-relevant hardware and firmware are functioning correctly and **FPT_TST_CIMC.2 (Software/firmware integrity test)** which covers the requirement to ensure that security-relevant software is functioning correctly.

O.Trusted Path is provided by **FTP_TRP.1 (Trusted path)** which covers the requirement that a trusted path between the user and the system be provided.

Security Objectives for the TOE and Environment

O.Configuration Management is provided by **FMT_MOF.1 (Management of security functions behavior) (iterations 1 and 2)** which covers the requirement that only authorized users can change the configuration of the system. **FMT_MOF_CIMC.3 (Extended certificate profile management)** covers the requirement that Administrators be able to control the types of information that are included in generated certificates. **FMT_MOF_CIMC.5 (Extended certificate revocation list profile management)** covers the requirement that Administrators be able to control to the types of information that are included in generated certificate revocation lists. **FMT_MOF_CIMC.6 (OCSP Profile Management)** covers the requirement that Administrators be able to control to the types of information that are included in generated OCSP responses. **O.Configuration Management** is supported by **AGD_ADM.1 (Administrator Guidance)** which covers the requirement that Administrators be provided with documentation describing the configuration management features of the TOE and by **A.Competent Administrators, Operators, Officers and Auditors** and **A.CPS** which ensure that Administrators are competent and are familiar with the CPS under which the TOE is to be operated. **O.Configuration Management** is also supported by **ACM_AUT.1 (Partial CM automation)**, **ACM_CAP.4 (Generation support and acceptance procedures)**, and **ACM_SCP.2 (Problem tracking CM coverage)** which ensure that a configuration management system is implemented and used.

O.Data import/export is provided by **FDP_UCT.1 (Basic data exchange confidentiality) (iterations 1 and 2)** and **FPT_ITC.1 (Inter-TSF confidentiality during transmission) (iterations 1 and 2)** which cover the requirement that data other than private and secret keys be protected when they are transmitted and from the CIMC. **FDP_ETC_CIMC.5 (Extended user private and secret key export)** and **FMT_MTD_CIMC.7 (Extended TSF private and secret key export)** cover the requirement that private and secret keys be protected when they are transmitted to and from the TOE.

O.Detect modifications of firmware, software, and backup data is provided by **FPT_TST_CIMC.2 (Software/firmware integrity test)** which covers the requirement that modifications to software or firmware be detected and **FDP_CIMC_BKP.2 (Extended CIMC backup and recovery)** which covers the requirement that modifications to backup data be detected. Since **FPT_TST_CIMC.2** and **FDP_CIMC_BKP.2** make use of digital signatures, keyed hashes, or authentication codes to detect modifications, **FMT_MTD_CIMC.4 (TSF private key confidentiality protection)** and **FMT_MTD_CIMC.5 (TSF secret key confidentiality protection)** are necessary to ensure that an attacker who has modified firmware, software, or backup data can not prevent detection of the modification by computing a new digital signature, keyed hash, or authentication code.

O.Individual accountability and audit records is provided by a combination of requirements. **FIA_UID.1 (Timing of identification) (iterations 1 and 2)** covers the requirement that users be identified before performing any security-relevant operations. **FAU_GEN.1 (Audit data generation) (iterations 1 and 2) and FAU_SEL.1 (Selective audit) (iterations 1 and 2)** cover the requirement that security-relevant events be audited while **FAU_GEN.2 (User identity association) (iterations 1 and 2)** and **FPT_STM.1 (Reliable time stamps) (iterations 1 and 2)** cover the requirement that the date and time of audited events are recorded in the audit records along with the identities of the entities responsible for the actions. **FMT_MTD.1 (Management of TSF data)** covers the requirement that audit data be available for review by ensuring that users, other than Auditors, can not delete audit logs. Finally, **FAU_SAR.1 (Audit review)** and **FAU_SAR.3 (Selectable audit review)** cover the requirement that the audit records are made available for review so that individuals can be held accountable for their actions.

O.Integrity protection of user data and software is provided by **FDP_ITT.1 (Basic internal transfer protection) (iterations 1 and 3)** and **FDP_SDI_CIMC.3 (Stored public key integrity monitoring and action)** which cover the requirement that user data be protected and **FPT_TST_CIMC.2 (Software/firmware integrity test)** and **FPT_TST_CIMC.3 (Software/firmware load test)** which cover the requirement that software and firmware be protected. Since data and software are protected using cryptography, **FMT_MTD_CIMC.4 (TSF private key confidentiality protection)** and **FMT_MTD_CIMC.5 (TSF secret key confidentiality protection)** are required to protect the confidentiality of the private and secret keys used to protect the data and software.

O.Limitation of administrative access is provided by **FDP_ACC.1 (Subset access control) (iterations 1 and 2)**, **FDP_ACF.1 (Security attribute based access control) (iterations 1 and 2)**, **FIA_UAU.1 (Timing of authentication) (iterations 1 and 2)**, and **FIA_UID.1 (Timing of identification) (iterations 1 and 2)**. **FIA_UAU.1 (Timing of authentication) (iterations 1 and 2)** and **FIA_UID.1 (Timing of identification) (iterations 1 and 2)** ensure that Administrators, Operators, Officers, and Auditors can not perform any security-relevant operations until they have been identified and authenticated and **FDP_ACC.1 (Subset access control) (iterations 1 and 2)** and **FDP_ACF.1 (Security attribute based access control) (iterations 1 and 2)** ensure that Administrators, Operators, Officers, and Auditors can only perform those operations necessary to perform their jobs. **FPT_RVM.1 Non-bypassability of the TSP (iteration 2)** ensure that Administrators, Operators, Officers, and Auditors can not perform operations that they are not authorized to perform by bypassing the TSP enforcement functions.

O.Maintain user attributes is provided by **FIA_ATD.1 (User attribute definition)** and **FIA_USB.1 (User-subject binding) (iterations 1 and 2)** which cover the requirement to maintain a set of security attributes associated with individual users and/or subjects acting on users' behalves. **FMT_MSA.1 (Management of security attributes)** ensures that only authorized users can modify security attributes. **FMT_SMF.1 (Specification of Management Functions)** ensures that and specific management functions are provided, like the management of users and permissions of access on the part of the users, and the administration of useres authentication.

O.Manage behavior of security functions is provided by **FMT_MOF.1 (Management of security functions behavior) (iterations 1 and 2)** which covers the requirement that authorized users be able to configure, operate, and maintain the security mechanisms. **FMT_SMF.1 (Specification of Management Functions)** ensures that and specific management functions are provided, like the management of users and

permissions of access on the part of the users, and the administration of useres authentication.

O.Object and data recovery free from malicious code is provided by **FPT_TST_CIMC.2 (Software/firmware integrity test)** and **FPT_TST_CIMC.3 (Software/firmware load test)** which cover the requirement that the recovered state is free from malicious code. **FDP_CIMC_BKP.1 (CIMC backup and recovery)**, and **FDP_CIMC_BKP.2 (Extended CIMC backup and recovery)** cover the requirement to be able to recover to a viable state.

O.Procedures for preventing malicious code is provided by **FPT_TST_CIMC.2 (Software/firmware integrity test)** which ensures that only signed code can be executed and **AGD_ADM.1 (Administrator Guidance)**, **AGD_USR.1 (User Guidance)** and **A.Malicious Code Not Signed** which ensure that those who are capable of signing code do not to sign malicious code. It is also supported by **FDP_ACF_CIMC.2 (User private key confidentiality protection)**, **FDP_ACF_CIMC.3 (User secret key confidentiality protection)**, **FCS_CKM.4 (Cryptographic key destruction)** and **FCS_CKM_CIMC.5 (CIMC private and secret key zeroization)** which ensure that an untrusted entity can not use a trusted entity's key to sign malicious code.

O.Protect stored audit records is provided by **FAU_STG.1 (Protected audit trail storage) (iterations 1 and 2)** which covers the requirement that audit records be protected against modification or unauthorized deletion and **FMT_MTD.1 (Management of TSF data)** which covers the requirement that audit records be protected from unauthorized access. **FPT_CIMC_TSP.1 (Audit log signing event)** is required so that modifications to the audit logs can be detected. **A.Physical Protection** is also required in order to protect the audit records from unauthorized physical modification. **FPT_ACC.1 (Access Control)** is also required in order to protect the audit records from unauthorized modification providing from any external program with access to the audit database.

O.Protect user and TSF data during internal transfer is provided by **FDP_ITT.1 (Basic internal transfer protection) (iterations 1-4)** which covers the requirement that user data be protected during internal transfer and **FPT_ITT.1 (Basic internal TSF data transfer protection) (iterations 1-4)** which covers the requirement that TSF data be protected during internal transfer.

O.Require inspection for downloads is provided by **FPT_TST_CIMC.3 (Software/firmware load test)** which covers the requirement that downloaded software can not be loaded until it has been signed and by **AGD_ADM.1 (Administrator Guidance)**, **AGD_USR.1 (User Guidance)**, and **A.Malicious Code Not Signed** which ensure that those who are capable of signing code do not to sign malicious code.

O.Respond to possible loss of stored audit records is provided by **FAU_STG.4 (Prevention of audit data loss) (iterations 1 and 2)** which covers the requirement that no auditable events, except those taken by the Auditor, can be performed when audit trail storage is full.

O.Restrict actions before authentication is provided by **FIA_UAU.1 (Timing of authentication) (iterations 1 and 2)** which covers the requirement that no security-relevant actions are performed on behalf of a user until that user has been authenticated.

O.Security-relevant configuration management is provided by **FMT_MSA.3 (Static attribute initialisation)** and **FMT_MSA.2 (Secure security attributes)** which cover the requirement that security attributes have secure values. **FMT_MOF.1 (Management of**

security functions behavior) (iterations 1 and 2) ensures that security-relevant configuration data can only be modified by those who are authorized to do so. **O.Security-relevant configuration management** is also supported by **AGD_ADM.1 (Administrator Guidance)** which covers the requirement that Administrators be provided with documentation describing the configuration management features of the TOE and by **A.CompetentAdministrators, Operators, Officers and Auditors** and **A.CPS** which ensure that Administrators are competent and are familiar with the CPS under which the TOE is to be operated.

O.Time stamps is provided by **FPT_STM.1 (Reliable time stamps) (iterations 1 and 2)** which covers the requirement that the time stamps be reliable, and by **A.NTP Client** which ensures this reliability by means the guarantee that all the hosts included in the TOE have installed an NTP client that synchronises the system clock with a reliable clock that obtains the Coordinated Universal Time from a reliable source.

O.User authorization management is provided by **FMT_MSA.1 (Management of security attributes)** which covers the requirement that Administrators manage and update user's security attributes**. O.User authorization management** is also supported by **AGD_ADM.1 (Administrator Guidance)** which covers the requirement that Administrators be provided with documentation describing the user authorization management features of the TOE and by **A.Competent Administrators, Operators, Officers and Auditors** and **A.CPS** which ensure that Administrators are competent and are familiar with the CPS under which the TOE is to be operated.

O.React to detected attacks is provided by **FCS_CKM.4 (Cryptographic key destruction)** and **FCS_CKM_CIMC.5 (CIMC private and secret key zeroization)** which cover the requirement that the user who detected the attack be able to destroy any plaintext keys within the TOE in order to prevent the attacker from obtaining copies of these keys. **FIA_AFL.1 (Authentication failure handling)** covers the requirement that the TSF respond to detected attacks (in the form of repeated authentication attempts) by taking actions to prevent the attacker from successfully authenticating him/herself. In the case that an attack is detected by an Administrator, Auditor, Officer, or Operator.

8.2.3 Rationale for operations of Security Requirements

This section contains justifications of some operations (assignments, selections, …) that has been applied to security requirements of the TOE or of the environment.

8.2.3.1 Rationale for operations of Security Requirements applied to the TOE

8.2.3.1.1 FIA_UAU.1.1

In this requirement the following actions have been included as events that are not security relevant:

- Indication of the authentication mode

  The following user authentication modes can be indicated:

  - Certificate (proof of possession using the private key related to the certificate that will be presented)

- Password
- Recovery password

Because the introduction of the indication of one of these authentication modes does not modify any security parameter of the system, but only it is an input parameter for the FIA_UIDAUT function (depending on this parameter the application will request the correct type of authentication data), then this event is not security relevant.

- Introduction of the authentication data

  These data can consist in the following type of information: password or the proof of possession generated using the private key of the indicated certificate. The introduction of these data does not modify any security parameter of the system, but only it is an input parameter for the FIA_UIDAUT function. For security reasons, characters introduced in the password field are presented on screen as asteriks, and the proof of possession is generated inside the smartcard that contains the private key related to the presented certificate. Consequently, this event is not security relevant.

- Cancel the login procedure

  The cancellation of the login procedure returns the state of the application to the previous state, and it eliminates any information introduced during the login process. Consequently, this event is not security relevant.

8.2.3.1.2 FIA_UID.1.1

In this requirement the following actions have been included as events that are not security relevant:

- Indication of the identification mode

  The following user identification modes can be indicated:

  - Certificate (presented in an authentication token, as an smart card)
  - Username
  - Recovery password

  Because the introduction of the indication of one of these identification modes does not modify any security parameter of the system, but only it is an input parameter for the FIA_UIDAUT function (depending on this parameter the application will request the correct type of identification data), then this event is not security relevant.

- Introduction of the identification data

  These data can consist in the following type of information: username or the indicated certificate. The introduction of these data does not modify any security parameter of the system, but only it is an input parameter for the FIA_UIDAUT function. Both the username and public key certificate can be considered not sensitive data in the system. Consequently, this event is not security relevant.

- Cancel the login procedure

The cancellation of the login procedure returns the state of the application to the previous state, and it eliminates any information introduced during the login process. Consequently, this event is not security relevant.

8.2.3.1.3 FDP_SDI_CIMC.3.2

In this requirement the following action has been included as event that is generated if the verification carried out to protect a public key fails: Generation of a report and forbid the use of the public key.

This action is consistent with the maintenance of the security, because:

- A report is generated and this guarantees that the system can monitor this security-relevant event in order it could be reviewed by an auditor for identifying the causes of the verification failure.
- The system forbids the use of the public key, and this guarantees that the integrity protection of these data is preserved.

8.2.3.1.4 FAU_STG.1.1

The TSF protects the stored audit records from unauthorized deletion because the KTS does not have any functionality to delete records from the audit database. From the KeyOne applications, it is not possibe to delete any registry from any database managed by these applications.

8.2.3.2 Rationale for operations of Security Requirements applied to the environment

8.2.3.2.1 FIA_UAU.1.1, FIA_UID.1.1

In these requirements the following action has been included as event that is not security relevant: Request for username and password.

The introduction of these data does not modify any security parameter of the system, but only they are input parameters for the identification and authentication function. For security reasons, characters introduced in the password field are presented on screen as hidden characters, and the username can be considered not sensitive data in the system. Consequently, this event is not security relevant.

8.2.3.2.2 FPT_TST_CIMC.2.2

In this requirement the following action has been included in the assignment operation: *report the test failure*.

This completion is consistent with maintenance of security because the inclusion of the action "report the test failure" in this requirement maintains the satisfaction of the the following security objectives by the FPT_TST_CIMC.2.2 requirement:

- O.Detect modifications of firmware, software, and backup data. The FPT_TST_CIMC.2.2 covers the requirement that modification to software or firmware be detected. This detection is possible by means the report that is generated after the test failure at power-up or on-demand.

- O.Integrity protection of user data and software. The FPT_TST_CIMC.2.2 covers the requirement that integrity of software and firmware be protected by means the detection of integrity errors at power-up and on-demand. This integrity protection is provided by security mechanisms that report the test failure after a lack of integrity.

- O.Object and data recovery free from malicious code. The FPT_TST_CIMC.2.2 requirement assures that the system stays free from malicious code since this requirement guarantees that any intrusion to the system is reported indicating a test failure.

- O.Periodically check integrity. The FPT_TST_CIMC.2.2 covers the requirement to periodically check the integrity of software (at power-up or on-demand). This check generates the evidence of a report indicating the test failure.

- O.Procedures for preventing malicious code. The FPT_TST_CIMC.2.2 ensures that only authenticated code (by means error detection code, authentication code, keyed hash or digital signature) can be executed. A report indicating a test failure, after power-up or on-demand, guarantees that the system is prevented against malicious code.

- O.Validation of security function. The FPT_TST_CIMC.2.2 ensures that only authenticated software and firmware is functioning, and therefore that they work correctly through features and procedures. The generation of a report indicating a test failure contributes guarantees in the maintenance of this security objective.

8.2.3.2.3 FPT_TST_CIMC.3.2

In this requirement the following action has been included in the assignment operation: *does not allow the execution of the component where the test has failed*.

This completion is consistent with maintenance of security because the inclusion of the action "does not allow the execution of the component where the test has failed" in this requirement maintains the satisfaction of the the following security objectives by the FPT_TST_CIMC.3.2 requirement:

- O.Integrity protection of user data and software. The FPT_TST_CIMC.3.2 covers the requirement that integrity of software and firmware be protected by means the detection of integrity errors at power-up and on-demand. This integrity protection is provided by security mechanisms that does not allow the execution of a software or firmware that is externally loaded into the system, where the test has failed.

- O.Object and data recovery free from malicious code. The FPT_TST_CIMC.3.2 requirement contributes in which the system remains free from malicious code since this requirement guarantees that no component (software or firmware) is executed if the integrity test has failed, when this component has been tried to load in the system.

- O.Periodically check integrity. The FPT_TST_CIMC.3.2 requirement contributes in the periodical check of the integrity of software, when a component (software or firmware) has been tried to load in the system, because if the test failed, then the system does not allow the execution of this component.

- O.Require inspection for downloads. The FPT_TST_CIMC.3.2 requirement contributes in the inspection for downloads, because this requirement guarantees that no component (software or firmware) is executed if the integrity test has failed, when this component has been tried to load in the system.

8.3 Internal Consistency and Mutual Support

This section demonstrates that the stated security requirements together form a mutually supportive and internally consistent whole. Internal consistency is semonstrated in an analysis of dependencies. Mutual support is shown through consideration of the interactions between and among the SFRs.

8.3.1 Rationale that Dependencies are Satisfied

The selected security requirements include related dependencies, both direct and indirect. The indirect dependencies are those required by the direct dependencies. All of these dependencies must be met or their exclusion justified.

8.3.1.1 Security Functional Requirements Dependencies

The following table provide a summary of the security functional requirements dependency analysis for this Security Target.

Component	Dependencies	Which is:
FAU_GEN.1 Audit data generation	FPT_STM.1 Reliable time stamps	Included
FAU_GEN.2 User identity association	FAU_GEN.1 Audit data generation	Included
	FIA_UID.1 Timing of identification	Included
FAU_SAR.1 Audit review	FAU_GEN.1 Audit data generation	Included
FAU_SAR.3 Selectable audit review	FAU_SAR.1 Audit review	Included
FAU_SEL.1 Selective Audit	FAU_GEN.1 Audit data generation	Included
	FMT_MTD.1 Management of TSF data	Included
FAU_STG.1 Protected audit trail storage	FAU_GEN.1 Audit data generation	Included
FAU_STG.4 Prevention of audit data loss	FAU_STG.1 Protected audit trail storage	Included

FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin	FIA_UID.1 Timing of identification	Included
FCO_NRO_CIMC.4 Advanced verification of origin	FCO_NRO_CIMC.3	Included
FCS_CKM.1 Cryptographic key generation	FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation	FCS_COP.1 Included
	FCS_CKM.4 Cryptographic key destruction	Included
	FMT_MSA.2 Secure security attributes	Included
FCS_CKM.4 Cryptographic key destruction	FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation	FCS_CKM.1 Included
	FMT_MSA.2 Secure security attributes	Included
FCS_CKM_CIMC.5 CIMC private and secret key zeroization	FCS_CKM.4 Cryptographic key destruction	Included
	FDP_ACF.1 Security attribute based access control	Included
FCS_COP.1 Cryptographic operation	FCS_CKM.4 Cryptographic key destruction	Included
	FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation	FCS_CKM.1 Included
	FMT_MSA.2 Secure security attributes	Included
FDP_ACC.1 Subset access control	FDP_ACF.1 Security attribute based access control	Included
FDP_ACF.1 Security attribute based access control	FDP_ACC.1 Subset access control	Included
	FMT_MSA.3 Static attribute initialization	Included
FDP_ACF_CIMC.2 User private key confidentiality protection	None	
FDP_ACF_CIMC.3 User secret key confidentiality protection	None	
FDP_CIMC_BKP.1 CIMC backup and recovery	FMT_MOF.1 Management of security functions behavior	Included
FDP_CIMC_BKP.2 Extended CIMC backup and recovery	FDP_CIMC_BKP.1 CIMC backup and recovery	Included

FDP_CIMC_CER.1 Certificate Generation	None	
FDP_CIMC_CRL.1 Certificate revocation list validation	None	
FDP_CIMC_CSE.1 Certificate status export	None	
FDP_CIMC_OCSP.1 OCSP basic response validation	None	
FDP_ETC_CIMC.5 Extended user private and secret key export	None	
FDP_ITT.1 Basic internal transfer protection	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_ACC.1 Included
FDP_SDI_CIMC.3 Stored public key integrity monitoring and action	None	
FDP_UCT.1 Basic data exchange confidentiality	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	Included
	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path	FTP_TRP.1 Included
FIA_AFL.1 Authentication failure handling	FIA_UAU.1 Timing of authentication	Included
FIA_ATD.1 User attribute definition	None	
FIA_UAU.1 Timing of authentication	FIA_UID.1 Timing of identification	Included
FIA_UID.1 Timing of Identification	None	
FIA_USB.1 User-subject binding	FIA_ATD.1 User attribute definition	Included
FMT_MOF.1 Management of security functions behavior	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
	FMT_SMF.1 Specification of management functions	Included
FMT_MOF_CIMC.3 Extended certificate profile management	FMT_MOF.1 Management of security functions behavior	Included
	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
FMT_MOF_CIMC.5 Extended certificate	FMT_MOF.1 Management of security functions behavior	Included

revocation list profile management	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
FMT_MOF_CIMC.6 OCSP profile management	FMT_MOF.1 Management of security functions behavior	Included
	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
FMT_MSA.1 Management of security attributes	FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control	Included
	FMT_SMF.1 Specification of management functions	Included
	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
FMT_MSA.2 Secure security attributes	ADV_SPM.1 Informal TOE security policy model	Included
	FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control	FDP_ACC.1 Included
	FMT_MSA.1 Management of security attributes	Included
	FMT_SMR.1 Security Roles	Included (hierarchical to FMT_SMR.2)
FMT_MSA.3 Static attribute initialization	FMT_MSA.1 Management of security attributes	Included
	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
FMT_MTD.1 Management of TSF data	FMT_SMR.1 Security roles	Included (hierarchical to FMT_SMR.2)
	FMT_SMF.1 Specification of management functions	Included
FMT_MTD_CIMC.4 TSF private key confidentiality protection	None	
FMT_MTD_CIMC.5 TSF secret key confidentiality protection	None	
FMT_MTD_CIMC.6 TSF private and secret key export	None	

FMT_MTD_CIMC.7 Extended TSF private and secret key export	FMT_MTD_CIMC.6	Included
FMT_SMR.2 Restrictions on security roles	FIA_UID.1 Timing of identification	Included
FPT_AMT.1 Abstract machine testing	None	
FPT_CIMC_TSP.1 Audit log signing event	FAU_GEN.1 Audit data generation	Included
	FMT_MOF.1 Management of security functions behavior	Included
FPT_ITC.1 Inter-TSF confidentiality during transmission	None	
FPT_ITT.1 Basic internal TSF data transfer protection	None	
FPT_STM.1 Reliable time stamps	None	
FPT_TST_CIMC.2 Software/firmware integrity test	FPT_AMT.1 Abstract machine testing	Included
FPT_TST_CIMC.3 Software/firmware load test	FPT_AMT.1 Abstract Machine Testing	Included
FTP_TRP.1 Trusted path	None	
FPT_ACC.1 Access Control	None	

Table 8-8. Summary of Security Functional Requirements Dependencies for Security Level 3

8.3.1.2 Security Assurance Requirements Dependencies

The following table provide a summary of the security assurance requirements dependency analysis for Security Level 3, with additional Security Assurance requirements to achieve a complete CC EAL4 Level.

Component	***Depends On:***	***Which is:***
ACM_AUT.1	ACM_CAP.3	Included (hierarchical to ACM_CAP.4)

	ALC_DVS.1	included
ACM_CAP.4	ALC_DVS.1	included
	ACM_CAP.3	included (hierarchical to ACM_CAP.4)
ACM_SCP.2	(indirect) ALC_DVS.1	included
	ACM_CAP.3	included (hierarchical to ACM_CAP.4)
ADO_DEL.2	(indirect) ALC_DVS.1	included
	AGD_ADM.1	included
	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
ADO_IGS.1	(indirect) ADV_RCR.1	included
	ADV_RCR.1	included
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
ADV_FSP.2	ADV_RCR.1	included
ADV_HLD.2	ADV_LLD.1	included
	ADV_RCR.1	included
ADV_IMP.1	ALC_TAT.1	included
	(indirect) ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
	(indirect) ADV_HLD.2	included
	ADV_HLD.2	included
	ADV_RCR.1	included
ADV_LLD.1	(indirect) ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
	no dependencies	not applicable
	ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
ADV_RCR.1	(indirect) ADV_RCR.1	included
ADV_SPM.1	ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
	(indirect) ADV_RCR.1	included

AGD_ADM.1	ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
	(indirect) ADV_RCR.1	included
AGD_USR.1	No dependencies	Not applicable
	No dependencies	Not applicable
ALC_DVS.1	No dependencies	Not applicable
ALC_FLR.2	ADV_IMP.1	Included
ALC_LCD.1	(indirect) ADV_FSP.1	Included (hierarchical to ADV_FSP.2)
ALC_TAT.1	(indirect) ADV_HLD.2	included
	(indirect) ADV_LLD.1	included
	(indirect) ADV_RCR.1	included
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
	ATE_FUN.1	included
ATE_COV.2	(indirect) ADV_RCR.1	included
	ADV_HLD.1	included (hierarchical to ADV_HLD.2)
	ATE_FUN.1	included
ATE_DPT.1	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
	(indirect) ADV_RCR.1	included
	No dependencies	Not applicable
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
ATE_FUN.1	AGD_ADM.1	included
ATE_IND.2	AGD_USR.1	included
	ATE_FUN.1	included
	(indirect) ADV_RCR.1	included
	ADO_IGS.1	included
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
AVA_MSU.2	AGD_ADM.1	included

	AGD_USR.1	included
	(indirect) ADV_RCR.1	included
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
	ADV_HLD.1	included (hierarchical to ADV_HLD.2)
AVA_SOF.1	(indirect) ADV_RCR.1	included
	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
	ADV_HLD.2	included
AVA_VLA.2	ADV_IMP.1	included
	ADV_LLD.1	included
	AGD_ADM.1	included
	AGD_USR.1	included
	(indirect) ADV_RCR.1	included
	(indirect) ALC_TAT.1	included

Table 8-9. Summary of Security AssuranceRequirements Dependencies for Security Level 3

8.3.2 Rationale that Requirements are Mutually Supportive

The requirements represented in this ST were developed from a variety of sources. The security work mutually so that each SFR is protected against bypassing, tampering, deactivation, and detection attacks by other SFRs.

Bypass

Prevention of bypass is derived as described below:

FIA_UID.1 and FIA_UAU.1 support other functions' allowing user access to data by limiting the actions the user can take prior to identification and authentication.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1 support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FPT_TST_CIMC.2 provides for integrity testing to ensure that selected security functions are operational, thus checking for bypass.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, thus providing protection from bypass to those SFRs dependent on that data.

Tamper

Prevention of tamper is derived as described below:

FAU_STG.1 protects the integrity of the audit trail.

FCS_CKM.1 and FCS_COP.1 provide for the secure generation and handling of keys, and therefore support those SFRs that may rely on the use of those keys.

FIA_UID.1 and FIA_UAU.1 support other functions allowing user access to data by limiting the actions the user can take prior to identification and authentication.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1 support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FPT_TST_CIMC.2 provides for integrity testing to ensure that selected security functions are operational, thus checking for tampering.

FDP_ETC_CIMC.5) prevent modification errors during export of secret and/or private keys.

FIA_AFL.1 supports all SFRs dealing with authentication by limiting the number of entry attempts, and then mandating an appropriate action to protect the TOE if too many attempts have been made.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, thus providing protection from tampering to those SFRs dependent on that data.

Deactivation

Prevention of deactivation is derived as described below:

The access control SFP detailed in FDP_ACF.1 along with the other SFRs dealing with access control, provide for rigorous control of allowed data manipulations and thus prevent unauthorized deactivation.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1, support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FPT_TST_CIMC.2 provides for integrity testing to ensure that selected security functions are operational, thus checking for tampering.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, thus providing protection from deactivation to those SFRs dependent on that data.

Detection

Detection is derived as described below:

The security audit functions, including FAU_GEN.1, FAU_GEN.2, and FAU_SEL.1 provide for the generation of audit data that may be used to detect attempts to defeat specific SFRs or potential misconfiguration that could leave the TOE prone to attack.

FAU_SAR.1 and FAU_SAR.3, support the audit generation SFRs by providing the capability to selectively search the audit records.

FAU_STG.1, and FAU_STG.4 provide for the protection of the audit records.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1, support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FMT_MSA.2 and FMT_MSA.3 limit the acceptable values for secure data, thus providing detection protection to those SFRs dependent on that data.

FMT_SMR.2 provides for the specification of multiple roles, thus supporting the other detection SFRs.

8.4 Rationale for Strength of Function

The TOE described in this ST is intended to operate in a range of environments, from benign to hostile.

Also, the users may be hostile. Therefore, the TOE requires cryptographic functions to provide for integrity, confidentiality, nondisclosure, and authentication. The authentication strength of function metrics provide for a basic level, and are currently within commercially available products. The cryptographic functions must be included in a cryptographic module that has been validated against FIPS 140-1 *Security Requirements for Cryptographic Modules*. The level required for the cryptographic module depends on the type and use of the key and the CIMC Security Level. The cryptographic module levels are specified in Table 6-5. FIPS 140-1 Level for Validated Cryptographic Module. The increasing FIPS 140-1 level corresponding to the increased CIMC Security Level addresses the increased threats and potential for loss at the higher levels.

The security and assurance requirements specified at CIMC Security Level 3 are intended for environments where the risks and consequences of data disclosure and loss of data integrity are moderate. CIMC Level 3 requires additional integrity controls to ensure data is not modified. CIMC Security Level 3 includes mechanisms to protect against attacks by parties with physical access to the components and includes additional assurance requirements to ensure the CIMC is functioning securely. The EAL for Security Level is EAL3 augmented, as defines [CIMC].

This TOE (KeyOne system) has been designed for accomplishing with Common Criteria EAL4+, so and as it is declared in this Security Target, where users requiere a moderate to high level of security.

8.5 Assurance Requirements Rationale

8.5.1 Rationale for CIMC security level 3

CIMC is designed to meet Security Level 3 may be appropriate for environments where risks and consequences of data disclosure and loss of data integrity are moderate. Level 3 requires additional integrity controls to ensure data is not modified. A CIMC at Security Level 3 includes protections to protect against someone with physical access to the components and includes additional assurance requirements to ensure the CIMC is functioning securely.

The assurance level for this Security Level is EAL 3/EAL 4 augmented. Augmentation results from the selection of:

ACM_SCP.2 Problem tracking configuration management coverage

A vendor can be expected to apply configuration management to the items called out in ACM_SCP.2.

Specifically, since the product is security related, the tracking of security flaws is a very reasonable expectation and within the bounds of standard, best commercial practice.

ADO_DEL.2 Detection of modification

A vendor can be expected to use a signature or other method to ensure that the code has not been tampered with prior to installation. Since the product is security related, this type of precaution should be expected.

ADV_FSP.2 Fully defined external interfaces

It is not a difficult task to fully define all external interfaces to the product. Indeed, this is necessary to correctly develop the product for interaction with other products. This will provide the necessary detail for supporting both thorough testing of the TOE and the assessment of vulnerabilities.

ADV_IMP.1 Subset of the implementation of the TSF

This high a level of assurance requires that additional documentation regarding the implementation of the product be provided. It is through examination of this portion of the implementation that the product can be adequately evaluated with regard to the requirements.

ADV_LLD.1 Descriptive low-level design

This high a level of assurance requires that additional documentation regarding the design of the product be provided. It is through examination of this design that the product can be adequately evaluated with regard to the requirements.

ADV_SPM.1 Informal TOE security policy model

While the generation of a security policy does require security expertise, this can be performed by a consultant (if necessary) and does not otherwise impact the vendor's existing development process at this Security Level.

ALC_FLR.2 Flaw Report Procedures

EAL 3 and EAL 4 do not have the ALC_FLR component. It is within best commercial practices for a vendor of security products to have flaw reporting procedures covering:

- Addressing user reported problems

- Correcting flaws

- Notifying users and

- Revising procedures to reduce the potential for introducing new and/or additional flaws.

Specific procedures are not defined in the assurance requirement, therefore this should have minimal impact on vendors who have already implemented a flaw reporting program.

ALC_TAT.1 Well-defined development tools

It is important that very secure products be unambiguous.

AVA_MSU.2 Validation of analysis components

A security vendor implementing standard, best commercial practices will not be impacted by this component. AVA_MSU.2 requires that the vendor produce user and administrator documentation that is adequate for understanding the operating modes of the TOE and the required external security controls necessary for secure operation. The vendor is required to analyze this documentation for conformance to the requirements.

AVA_VLA.2 Independent vulnerability analysis

Penetration attacks are very likely given the threat model for this Security Level. As a result, it is important that some penetration analysis and testing be performed.

8.5.2 Rationale for EAL4

The assurance requirements defined for security level 3 of PP CIMC are very nearly from CC EAL4, so EAL4 has been selected to be the overall assurance for this TOE. Additional assurance requirements are rationalized below:

ACM_AUT.1 Partial CM automation

Automation in the configuration management system can help reduce the risk of human error or negligence.

ACM_CAP.4 Generation support and acceptance procedures

It is important that changes to the TOE be appropriately controlled. This requirement helps to ensure that when ¡changes are made, they are appropriate and correctly applied to the resulting TOE.

ALC_LCD.1 Developer defined life-cycle model

It is important that changes to the TOE be appropriately controlled. This requirement helps to ensure that the development and maintained are appropriately controlled.

8.6 Rationale for the propietary extended security requirements

This Security Target specifies the following two types of extended security requirements:

- CIMC extended security requirements

  These requirements are included in the CIMC Extended Security Functional Requirements section, page 91, and they are not justified because these requirements are included in the [CIMC] Protection Profile.

- Propietary extended security requirements

  These requirements are propietary requirements of this Security Target. The following section describes them.

8.6.1 Propietary extended security requirements

Access Control (FPT_ACC)

Family Behaviour

This family defines requirements about the access control to the tools and programs that can be available by the TOE.

Component Leveling

FPT_ACC: Access Control to the software	1

At FPT_ACC.1, the TSF shall introduce requirements necessary to include access control to any software that could be available by the TOE.

Audit: FPT_ACC.1

There are no auditable events foreseen.

FPT_ACC.1 Access Control to the software

This component requires access control measures to be applied to those software that can be available by the TOE.

FPT_ACC.1.1

The environment must not have installed any [assignment: *software component*] that access to the [assignment: *technological component*] used by the TOE.

FPT_ACC.1.2

If [assignment: *software component*] that access to the [assignment: *technological component*] are used, then this access must be controlled and supervised by the [assignment: *role*].

9 Bibliography, Definitions and Acronyms

9.1 Bibliography

The following documents are referenced in this document:

Reference	*Referenced document*
[CEN01a]	CEN/ISSS Workshop on Electronic Signatures. CEN *Hardware Security Modules for CSPs, CC Protection Profile*, EESSI Area D2, 2001.
[CEN01b]	CEN/ISSS Workshop on Electronic Signatures. *CEN/ISSS WS/E-Sign Workshop Agreement Group F, Security Requirements of Secure Signature Creation Devices (SSCD)*, 2001.
[CEN01c]	CEN/ISSS Workshop on Electronic Signatures. *Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures*, June 2003.
[CIMC]	*Certificate Issuing and Management Components Family of Protection Profiles*, Version 1.0. October 31, 2001. National Security Agency (NSA).
[Eur99a]	European Community. *Algorithms and parameters for algorithms, list of algorithms and parameters eligible for electronic signatures, procedures as defined in the directive 1999/93/EC, article 9 on the "Electronic Signature Committee" in the Directive*., 1999.
[Eur99b]	European Community. *Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a community framework for electronic signatures*, 1999.
[FIP]	*FIPS 140-2 SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES.*
[Ser97]	Service Central de la Sécurité des Systèmes d'Information. *Expression des Besoins et Identification des Objectifs de Sécurité*, 1.02 edition, 1997.

Reference	Referenced document
[the99a]	the Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation* Part 3: Security assurance requirements, 2.2, January 2004.
[the99b]	the Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements*, 2.2, January 2004.
[the99c]	the Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model*, 2.2, January 2004.
[TS1]	ETSI TS 101 456, *Policy Requirements for Certification Authorities Issuing Qualified Certificates*.
[FUNCSPEC]	KeyOne 3.0 – Product Specification, Safelayer internal code: 6D6436D9
[ALGO]	ETSI SR 002 176 – Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures
[X509]	X509v3: ITU-T Recommendation X.509 | ISO/IEC International Standard 9594-8, Information Technology – Open Systems Interconnection – The Directory: Authentication Framework
[TS101862]	ETSI TS 1O1 862, Qualified Certificate Profile
[PKCS5]	PKCS #5: Password-Based Encryption Standard, RSA Laboratories
[RFC2560]	RFC 2560: Online Certificate Status Protocol - OCSP
[RFC3161]	RFC 3161: Time-Stamp Protocol (TSP)
[Eur03c]	*COMMISSION DECISION of 14 July 2003 on the publication of reference numbers of generally recognised standards for electronic signature products in accordance with Directive 1999/93/EC of the European Parliament and of the Council*
[CONFGUIDE]	Configuration Guide – CC EAL4 Certification -, Safelayer internal code: 8ADC23DA
[NPKI]	NATO Public Key Infrastructure (NPKI) Certificate Policy

9.2 Definitions

Activation data: Data values, other than keys, that are required to operate cryptographic modules and that need to be protected (e.g., a PIN, a passphrase, or a manually-held key share).

Advanced electronic signature: an electronic signature which meets the following requirements:

- it is uniquely linked to the signatory;
- it is capable of identifying the signatory;
- it is created using means that the signatory can maintain under his sole control; and
- it is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable;

CA-certificate: A certificate for one CA issued by another CA.

CRL distribution point: A directory entry or other distribution source for CRLs; a CRL distributed through a CRL distribution point may contain revocation entries for only a subset of the full set of certificates issued by one CA or may contain revocation entries for multiple CAs.

Certificate Dissemination Service: A service that disseminates certificates to Subscribers, and if the subscriber consents, to Relying Parties. This service also disseminates the CA.s policy & practice information to Subscribers and Relying Parties.

Certificate Generation Service: A service that creates and sign certificates based on the identity and other attributes verified by the registration service.

Certificate policy: A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.

Certificate validity period: The certificate validity period is the time interval during which the CA warrants that it will maintain information about the status of the certificate.

Certificate: an electronic attestation that links signature-verification data to a person and confirms the identity of that person;

Certificate: the public key of a user, together with some other information, rendered unforgeable by encipherment with the private key of the certification authority which issued it.

Certificate Issuing Management Component (CIMC): A Certificate Issuing Management Component consists of the hardware, software, and firmware that are responsible for performing the functions of a Certificate Issuing Management System. A CIMC does not include the environmental controls (e.g., controlled access facility, temperature), policies and procedures, personnel controls (e.g., background checks and security clearances), and other administrative controls that complete a CIMS.

Certification Practice Statement: A statement of the practices that a Certification Authority employs in issuing certificates.

Certification authority (CA): An authority trusted by one or more users to create and assign certificates. Optionally the certification authority may create the users keys.

Certification path: A chain of multiple certificates, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs.

Certification-service-provider: an entity or a legal or natural person who issues certificates or provides other services related to electronic signatures;

Digital signature: Data appended to, or a cryptographic transformation (see cryptography) of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery e.g. by the recipient.

Electronic signature: data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication of that data.

Electronic-signature-product: hardware or software, or relevant components thereof, which are intended to be used by a certification-service-provider for the provision of electronic-signature services or are intended to be used for the creation or verification of electronic signatures.

End entity: A certificate subject that uses its public key for purposes other than signing certificates.

Hash function: A function that maps string of bits to fixed-length strings of bits, satisfying the following two properties:

- It is computationally infeasible to find for a given output an input that maps to this output.
- It is computationally infeasible to find for a given input a second input which maps to the same output.

Policy qualifier: Policy-dependent information that accompanies a certificate policy identifier in an X.509 certificate.

Private key: That key of an entity's asymmetric key pair that should only be used by that entity.

Public key: That key of an entity's asymmetric key pair that can be made public.

Qualified certificate: a certificate that meets the requirements laid down in Annex I of the Directive and is provided by a certification-service-provider who fulfils the requirements laid down in Annex II of the Directive;

Qualified electronic signature: an advanced electronic signature which is based on a qualified certificate and which is created by a secure-signature-creation device (Note: Definition of 5.1 signature taken from the Directive)

Registration Service: A service that verifies the identity and, if applicable, any specific attributes of a Subscriber. The results of this service are passed to the Certificate Generation Service.

Registration authority (RA): An entity that is responsible for identification and authentication of certificate subjects, but that does not sign or issue certificates (i.e., an RA is delegated certain tasks on behalf of a CA).

Relying party: A user or agent that relies on the data in a certificate in making decisions.

Revocation Management Service: A service that processes requests and reports relating to revocation to determine the necessary action to be taken. The results of this service are distributed through the Revocation Status Service.

Revocation Status Service: A service that provides certificate revocation status information to relying parties. This service may be a real-time service or may be based on revocation status information that is updated at regular intervals.

Secure-signature-creation device: a signature-creation device that meets the requirements laid down in Annex III of the Directive;

Security policy: The set of rules lay down by the security authority governing the use and provision of security services and facilities.

Self-signed certificate: A certificate for one CA signed by that CA.

Signatory: a person who holds signature-creation data and acts either on his own behalf or on behalf of the natural or legal person or entity he represents; Note: the term signer is sometimes used as a synonym.

Signature-creation data: unique data, such as codes or private cryptographic keys, which are used by the signatory to create an electronic signature;

Signature-creation device: configured software or hardware used to implement the signature- creation data.

Signature-verification device: configured software or hardware used to implement the signature-verification-data.

Signature-verification-data: data, such as codes or public cryptographic keys, which are used for the purpose of verifying an electronic signature.

Subscriber Device Provision Service: A service that prepares and provides a Signature Creation Device to Subscribers.

Subscriber: An entity subscribing with a CSP to have its public key and identity certified in a public key certificate.

Time Stamping Service: A service that provides a trusted association between a datum and a particular point in time, in order to establish reliable evidence indicating the time at which the datum existed.

Trustworthy system: An information system or product implemented as either hardware and/or software that produces reliable and authentic records that are protected against modification and additionally ensures the technical and cryptographic security of the processes supported by it.

Voluntary accreditation: any permission setting out rights and obligations specific to the provision of certification services, to be granted upon request by the certification service provider concerned, by the public or private body charged with the

elaboration of, and supervision of compliance with, such rights and obligations, where the certification-service-provider is not entitled to exercise the rights stemming from the permission until it has received the decision by the body.

Note: The term "accreditation" is generally used in another way, meaning "accreditation of certification bodies performing conformity assessment of products and/or services".

9.3 Acronyms

The following abbreviations are used in this document:

Acronym	*Meaning*
ARL	Authority Revocation List
CA	Certification Authority
CIMC	Certificate Issuing and Management Components
CP	Certificate Policy
CRL	Certificate Revocation List
CSP	Certification Service Provider
HSM	Hardware Security Module
HW	Hardware
I/O	Input/Output
It	Information Technology
LRA	Lightweight Registration Authority
NDCCP	Near Domain Cert-Status Coverage Protocol
NQC	Non-Qualified Certificate
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OS	Operating System
PKI	Public key Infrastructure
POP	Proof Of Possession
PP	Protection Profile
QC	Qualified Certificate
RA	Registration Authority

Acronym	*Meaning*
SCD	Signature-Creation Device
SF	Security Function
SSCD	Security Signature Creation Device
ST	Security Target
TOE	Target Of Evaluation
TSA	Time Stamping Authority
TSP	Time-Stamp Protocol
TST	Time Stamp Token
TSS	Time Stamping Service
KTS	KeyOne Trustworthy System
VA	Validation Authority

APPENDIX A

Considerations about the license file

In order to install Safelayer products found in the distribution CD-Rom, a license file is necessary (`.sly`). In the license file a *cust* file is included that allows the execution of the Safelayer applications.

The *cust* file is accessed by the Safelayer software to verify the compliment of the licenses during the execution of the applications. This *cust* is installed during the set-up procedure in the appropriate location. Safelayer applications will not start up if this file does not exist or if the data contained in this file do not allow the execution of the application.

Safelayer applications, when started, validate all scripts that the product is going to use. If during this process a script is found with a non-recognized or invalid signature, the application shows an integrity error in scripts and it stops.

In the *cust* file root certificates are included, necessary for the verifying code signatures. Normally, two root certificates are included:

- The first one corresponds to the Safelayer signing code, necessary for verifying the code contained in the distribution CD-Rom. This certificate is included initially in the first cust delivered from Safelayer.
- The second one is individual for each installation. This certificate allows to verify that the client has signed changes that have been performed in the Safelayer product scripts, preventing that these changes be performed by non-authorized personnel. This certificate will be included in the *cust* by Safelayer in order to sign any script.

Each time a KeyOne application is started up, the signature of all of its scripts is verified. If any script has an invalid signature, an error message stops the application from being used. Depending on the license file, it can allow the execution of scripts without validate the signature related to it. This is possible if this license file allow execute the scripts with the flag `--unsecure`.

In order to fulfill with the EAL4+ security guarantees of the product, the license file that is used does not allow the execution of the scripts with this flag. The way to check if this flag is allowed is to try any application or script with this flag. One example is to add the flag `--unsecure` in the starting file of the KeyOne CA server (ex. `start C:\Safelayer\KeyOne30\KeyOneHome\keyoneserver\keyoneserver.exe -configfile "./online/start_server.ws" --unsecure`).

If the license file allows the execution of scripts in unsecure mode, then the following error report will appear: "`Warning! KeyOne Server is running in unsecure mode.Scripts signature is not verified in any way`". Anyway, an error message similar to the following one stops the application to being used: "`Invalid program arguments. ERROR - Scryptor Event_ForbiddenParameter. Parameter = -- unsecure`".

SAFELAYER SECURE COMMUNICATIONS, S.A.

Edificio Valrealty C/ Basauri, 17 Edifico B Pl. Baja Izq. Of. B 28023 Madrid (SPAIN) Tel.: +34 91 7080480 Fax: +34 91 3076652
Edif. World Trade Center (S-4), Moll de Barcelona S/N 08039 Barcelona (SPAIN) Tel.: +34 93 5088090 Fax: +34 93 5088091

WWW.SAFELAYER.COM