Security Target

KeyOne 4.0

SAFELAYER

KeyOne 4.0 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 Security Target and TOE Reference

Document Identifier	**95A278AC v2.1**
Title	Security Target – KeyOne 4.0
Issue Date	September, 2014
Release Identifier	Release Base: 4.0.13S2R1 Release Patches: 4.0.13S2R1_B01, 4.0.13S2R1_B02
Authors	Safelayer Secure Communications S.A.
CC Version	Common Criteria version 3.1 Release 4
Evaluated TOE	KeyOne CA 4.0, KeyOne XRA 4.0, KeyOne VA 4.0, CRL Authority Add-in for KeyOne CA, ePassport Country Verifying CA Add-in for KeyOne CA and ePassport Document Verifier Add-in for KeyOne CA.
TOE Name	KeyOne CA 4.0, KeyOne XRA 4.0, KeyOne VA 4.0

From a commercial point of view, the products evaluated correspond to the "Evaluated TOE" row of the previous table, i.e., the following products: KeyOne CA 4.0, KeyOne XRA 4.0, KeyOne VA 4.0, the CRL Authority Add-in for KeyOne CA, the ePassport Country Verifying CA Add-in for KeyOne CA and the ePassport Document Verifier Add-in for KeyOne CA.

1.2 TOE Overview

1.2.1 KeyOne Certification Authority

KeyOne CA is a software application that performs the Certification Authority functions of issuing public key digital certificates using the syntax defined in ITU-T X.509v3.KeyOne CA forms part of the Safelayer Public Key Infrastructure (PKI) solution. KeyOne CA can act as a Root CA, Subordinate CA, Cross CA, Bridge CA, Online CA and Offline CA. Depending on how it is used, the CA operates in conjunction with a Registration Authority product that assumes the entity registration functions. KeyOne CA can also operate in conjunction with the Validation Authority product to provide the digital certificate validation service. The main functions of KeyOne CA are to:

- Generate and protect the private keys via the use of cryptographic devices (HSM).
- Automatically manage the life-cycle and the coexistence of the private keys of the CA.
- Manage recognized RAs and assign them certification policies.
- Generate the ITU-T X509v3 digital certificates (for users and applications) requested by the RAs.
- Generate and publish lists of revoked and suspended digital certificates (CRLs).
- Report on the status of the digital certificates so the validation service (VA) can publish it via OCSP.
- Guarantee the secure auditing of the events and actions carried out in the system.

KeyOne CA is designed to facilitate compliance with the security requirements for trustworthy systems managing certificates for electronic signatures (CEN CWA 14167-1) in terms of roles and events. It facilitates adaptation to the ETSI TS 101 456 recommendations for certification authority policies that issue recognized digital certificates. The system support FIPS 140-2 level 3 HSMs.

1.2.2 KeyOne Registration Authority

KeyOne XRA is part of the Safelayer Public Key Infrastructure (PKI) solution. KeyOne XRA operates as a user/application registration service (RA) for requesting the issuing and revocation of digital certificates (in conjunction with KeyOne CA).

KeyOne XRA is extremely adaptable to business needs: for user registration processes and for the delivery of digital certificates to users. Its workflow manager provides simple and reliable system configuration for defining what data processing actions are to be included in the registration process and what data the system is to exchange with users, operators and applications.

The main functions of KeyOne XRA are to:

- User registration and digital certificate life-cycle management through interaction with KeyOne CA.

- Certificate life-cycle management for PKI services and applications that require authentication, signature and data encryption.

KeyOne XRA includes the role management, auditing and reporting mechanisms recommended for digital certificate management systems for CEN CWA 14167-1 e-signature. It facilitates adaptation to the ETSI TS 101 456 recommendations for the policies of certification authority policies that issue recognized digital certificates. The system support FIPS 140-2 level 3 HSMs.

1.2.3 KeyOne Validation Authority

KeyOne VA is suitable for critical processes of electronic signature validation since it provides evidential value and greater efficiency in the verification of the status of the digital certificates (in contrast to the conventional mechanism which are based in revocation lists).

The main functions of KeyOne VA are to:

- Store information on the status of the certificates generated by one or more Certification Authorities. The status of a digital certificate is updated by downloading the revocation lists or the information provided by Certification Authorities (CA) that have the KeyOne publication service (KeyOne CertStatus Server) installed. In both cases, updating is performed remotely.

- Receive user or service-provider requests on the status of the digital certificates used in the signing of electronic transactions.

- Guarantee the non-repudiation of the responses. These responses are digitally-signed by the Validation Authority and specify the date and status (valid, revoked, cancelled or unknown) of a certificate.

- Generate event logs so operators can monitor the system status, its security and to what extent the corporate specifications are being met.

Customize the system to tailor response delivery and content to the identity of the requester.KeyOne products support defining the roles and events required to operate in compliance with the Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures (CWA14167-1). KeyOne VA supports the roles of security operator, system administrator and system auditor. The system support FIPS 140-2 level 3 HSMs.

1.2.4 Environment Components

The TOE relies upon the following IT additional hardware and software:

- Operating System: Windows Server 2012.

- Databases: Oracle 11g R2, Microsoft SQL Server 2012 Enterprise Edition SP1.

- Hardware Security Module: nCipher nShield Connect 1500, Safenet Luna SA 5.1.1 (FIPS 140-2 level 3).

- Java Runtime: JRE 1.7.0.

- Certificate-based PKI USB authenticator: Safenet eToken 510x.

1.3 TOE Description

1.3.1 Physical Scope of the TOE

The TOE physically consists of a DVD labeled "KeyOne <version identifier>", where < version identifier> is the identifier of the release[1].

The distribution DVD is supplied by Safelayer Secure Communications S.A. and contains the following components:

- Installation Application

  The Safelayer distribution DVD contains a graphical installation application for installing. uninstalling, checking and repairing KeyOne applications and for installing and uninstalling patches in KeyOne applications. It also contains the following guidance documentation: [K140CSO], [K140PIU], [K140UM] and [K140AU].

- Release Notes

  The distribution DVD includes a file with the following name: "KeyOne 4.0 - Release notes <releaseId>.pdf", where <releaseId> is the identifier of the release. This document describes the following aspects of the KeyOne installation: bugs fixed since the last release, functionality added since the last release and known issues in the current release.

- Keyfile

  The keyfile (with the .sly extension) includes a list of modules that can be installed, updated and uninstalled from DVD. KeyOne products come encrypted on the distribution DVD and can only be installed with the keyfile supplied by Safelayer.

- Java Runtime Environment

  The graphical applications of the KeyOne applications use Java technology. It is necessary to install the Java Runtime Environment before starting up the installation tool or any KeyOne application.

- Visual C++ Redistributable Package

  The installation of KeyOne applications in Windows platforms requires the previous installation Microsoft Visual C++ Redistributable Package.

[1] *According the information about the evaluated TOE included in section 1.1 of this document.*

1.3.2 Logical Scope of the TOE

In this section, the logical scope of the TOE is described. Other technical details about this logical scope are discussed in section 1.3.3 The TOE.

The logical scope of the TOE is described based on the following security functional requirements:

Security audit: Security requirements for logging the occurrence of security relevant events that take place under TOE control and creating and maintaining a secure audit trail. The TOE maintains a logging system that ensures the integrity of the recorded events.

Access control policy: The control access policy is defined following a role-based security approach and according to the specific security policy configured in the TOE.

Secure communications: The TOE provides integrity and confidentiality protection of user and system data when it is transferred between different components or products.

Identification and authentication: The TOE defines types of user identification and authentication mechanisms. These mechanisms may be restricted by the security policy depending on the level of security they provide.

Private and public key protection: The TOE ensures that private keys can be stored in a FIPS 140-2 validated cryptographic module which ensures the confidentiality and security of these keys. The TOE provides internal mechanisms and security protocols to ensure the integrity of the public keys.

Backup and recovery: The TOE provides backup and recovery mechanisms of the user and system data. When it restores the state of the system, the recovery function creates an “equivalent” system state in which information about all relevant TOE transactions is maintained. The backup data is protected against modification through the use of digital signatures.

Certificate status export: The TOE allows the export of the certificate status by means of the generation of X.509 Certification Revocation Lists. It also offers the possibility of using the standard OCSP protocol (RFC 2560) available in the KeyOne VA product. The certificate status is securely synchronized between KeyOne CA and KeyOne VA products.

Management of keys and certificates: The TOE generates X509 certificates and automatically manages the keys and certificates life-cycle in all its components.

Certificates and certificate revocation lists profiles: The TOE manages certificate and certificate revocation list profiles. A profile defines the set of acceptable values for fields and extensions in a certificate or in a certificate revocation list (CRL). The TOE requires the Administrator to specify the set of acceptable values for the fields and extensions of the certificate or CRL. The TOE ensures that issued certificates and CRLs are consistent with the defined profile.

1.3.2.1 PKI Technology

The Internet and electronic relations are the new way of interacting. Electronic mechanisms have automated transactions and made paper documents a thing of the past.

Public key technology manages the risks so as to guarantee the security of the electronic transactions carried out on open and unsecure networks, such as the Internet. PKI is fundamental in:

- Improving business processes as it optimizes times, manages errors and reduces costs.
- Improving client and user satisfaction as it makes possible interactions from anywhere at anytime.

1.3.2.1.1 Security Services

Electronic transactions are only possible when the persons and machines involved can be reliably identified. Public key technology is the most secure tool for electronic identification and data protection.

The public key infrastructure uses digital certificates (a sort of electronic identity card) to provide a set of services that greatly reduce the security risks associated with business processes. PKI provides:

- **Electronic identification** that univocally guarantees the identity and attributes of an entity. It answers the questions of who and what. The identity provides the name of a person or machine, whereas the attributes provide information on the entity, such as capacity to act as a qualified professional, credit limit or date of birth.
- **Data integrity** that detects any accidental or intentional change made to data in storage or being transmitted via telematic networks. The authentication and integrity services are the base for the **electronic signature**. As the electronic signature is equivalent to the hand-written one, paper becomes unnecessary.
- **Confidentiality** that protects electronic data (files and communications) and controls access to it using PKI-based authentication mechanisms.

The public key infrastructure (PKI) provides the services required for establishing trusted electronic relations. To ensure this, the Trusted Third Parties (TTP):

- Guarantee the univocal relation between entities and their socio-economic data.
- Univocally associate a date with specific data.
- Provide proof that already-established relations are still valid.

A type of Trusted Third Parties is the **Certification Authority** that is the responsible for issuing and managing digital certificates.

1.3.3 The TOE

The TOE associated with this Security Target is composed of the KeyOne CA, the KeyOne VA and the KeyOne XRA products.

KeyOne CA is an application for creating and managing Certification Authorities of various types (root/subordinate, online/offline), allowing full management lifecycle of user certificates.

The purpose of KeyOne CA is to implement the functionality of a X.509 Certification Authority X.509 and implement the required functionality of the CV certificates Certification Authorities to establish the **electronic passport**.

In the KeyOne PKI, KeyOne CA implements the certificate management system. This application:

- Provides services of X.509 and CV certificates generation, to issue certificates to subscribers.
- Defines certification profiles that apply to requests for X.509 certificates and CV processed by the certificate generation service.
- Provides a service of X.509 certificate revocation, to process incoming requests for revocation that may result in suspension, revocation or rehabilitation of certificates of subscribers.
- Provides a CV certificates revocation, to process requests for revocation that result in the impossibility of automatically authenticate a subsequent certification request with the key of the revoked certificate.
- Provides a service to generate X.509 CRLs, in order to issue the certificate revocation lists that can be exported and published in repositories to be accessed by entities that are interested in checking the status of certificates of subscribers.
- Defines the X.509 CRLs profiles that set fields and extensions of the CRLs that will be issued.
- Guarantees the secure auditing of the events and actions carried out in the system.

KeyOne CA implements public key certification functions (using the syntax defined in [ITU-T X.509v3]). These functions are accessible via:

- The graphical interface of the application.
- The SOAP/WS interface of the application server.
- Other KeyOne components, as the KeyOne XRA (that acts as the Registration Authority).

The functions of KeyOne CA can be extended with the KeyOne CRLA extension and a set of optional extensions for implementing electronic passport infrastructures.

In KeyOne, the generation of CRLs is implemented with the **KeyOne CRLA** extension for KeyOne CA. This extension periodically updates and generates CRLs that contain all the certificates revoked by a certification system at a point in time.

The **electronic passport** is divided into the following two components:

- The Extended Access Control (EAC) infrastructure for the second generation of e-Passports. In this case the passport is able to authenticate the reader using a PKI of CV certificates. Thus, the passport can restrict access from the data, depending on the access permissions contained in the certificate of the reader.
- The infrastructure for the first generation of electronic travel documents, which was standardized by the International Civil Aviation Organization (ICAO). In this case the data contained in the electronic passports are signed using a PKI of X.509 certificates. Thus the application that accesses this data can validate that it is a passport which data are integer (have been generated by a legitimate entity). Protection against cloning uses a system based on a challenge-response called *Active Authentication (AA)*.

KeyOne XRA is an application for implementing all the registration functions. It:

- Registers the data of end entities.
- Generates certification requests for end entities.
- Sends the certificates to the owners and publishes them in the repositories.
- Generates certificate renewal and revocation requests.

These functions are accessible via the graphical interface of the application.

KeyOne VA is a system for electronic-signature verification critical processes. The main advantages of this system are:

- Proof of response delivery.
- Greater efficiency in validating certificate status.

The KeyOne VA validation service is based on the IETF OCSP. The system:

- Responds to the requests for information on the status of digital certificates used in the signing of electronic transactions. These requests may come from users or service providers.
- Stores information on the status of certificates generated by one or more Certification Authorities.
- Guarantees the non-repudiation of the responses. The responses include a digital signature from the Validation Authority that specifies the date and status (valid, revoked, suspended or unknown) of the certificate.

1.3.3.1 Architecture

The PKI systems implemented by the KeyOne products are run in a shared environment known as the KeyOne system kernel, or, more simply, the KeyOne system. KeyOne applications share an execution and administration environment.

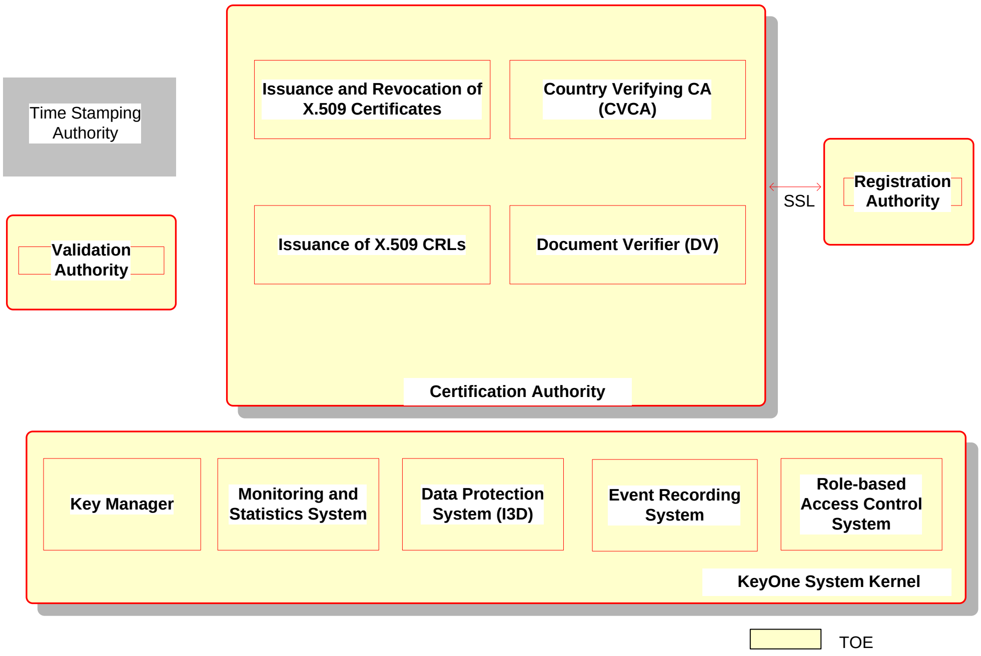


Figure 1-1. KeyOne CA Architecture

Using KeyOne CA it is possible to define the following three types of Certification Authorities:

Generic X.509 Certification Authority

X.509 Certification Authority that allows the issuance of X.509 certificates and CRLs. Depending on your settings can be a root or subordinate CA, and can provide its certification services online or offline through user interfaces that can be accessed by an operator.

Subscribers are individuals or entities to which the CA issues certificates. These subscribers can not access directly to services offered by KeyOne CA. Only the KeyOne CA operators can process certification or revocation requests through the user interface.

KeyOne CA allows online access to the interface of issuance and revocation of certificates to the approved Registration Authorities (RA). KeyOne XRA is a axample of Registration Authority that access to the online service.

An example of CA that can be implemented using KeyOne CA is the root Certification Authority needed in the BAC electronic passport, called *Country Signing Certification Authority (CSCA)*. In this case the *Document Signer*, that is the entity who signs the data contained in the passports, are the subscriptors of this CA.

Country Verifying CA (CVCA)

CV Root Certification Authority of the EAC ePassport. Each country creates its own CVCA, ie a hierarchy of CV certificates are generated by country. The CVCA of a country must issue certificates to all countries wishing to certify their passport readers so they can electronically access the data of their passports.

Entities to which the CA issues the certificates can not directly access to the CA services. Are the CA operators which can process certification and revocation requests through the CA user interface.

KeyOne CA allows online access to the interface of issuing certificates to the authorized Registration Authorities (RA). The KeyOne CVRA is an example of RA of this type of CA.

Document Verifier (DV)

CV subordinated Certification Authority of the EAC ePassport. This CA is certified by all countries, having to manage multiple sets of keys and the corresponding certificates, one for each country to which service.

This type of CA generates certificates for the electronic passport readers, which are managed by entities called Inspection System (IS). The passport reader must provide the certificate of the hierarchy of the country where the passport was issued, and therefore it must manage as many certificates as countries manage the DV.

Are the CA operators who can process certification and revocation requests, of the inspection systems through the CA graphical interface.

Inspection systems can access online to the CA interface of issuance of certificates.

To be able to automate the issue and renewal of certificates from the certification authority, an interface with the superior RA in the hierarchy, the Single Point of Contact (SPOC), is defined. The SPOC can communicate with all the CVCAs of all the countries for processing the certification requests from the DV. Because certificate issue time is indefinite, as, depending on the certificate request, foreign CVCAs must be contacted, communication is asynchronous and KeyOne CA provides an online interface for the SPOC to send the certificates when they are available.

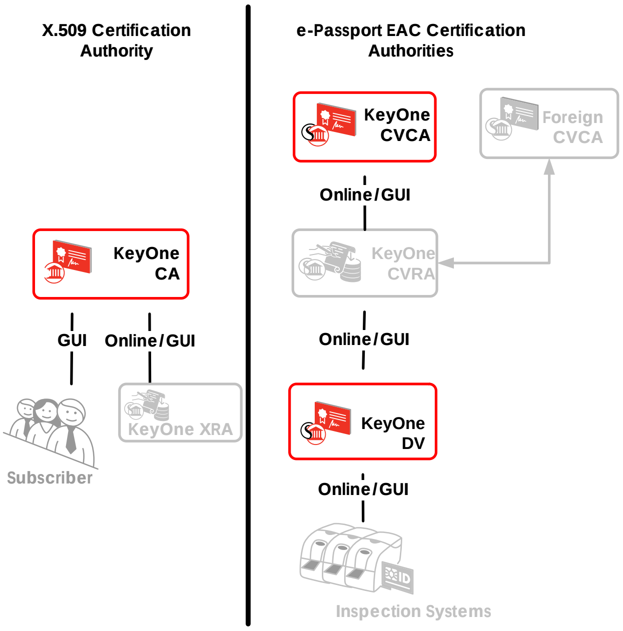


Figure 1-2. Types of Certification Authorities

Validation Authority (VA)

The VA issues evidences that certify the validity of digital certificates. To generate these evidences, a VA can access two complementary sources of information:

- A CA certification status service
- Certificate revocation lists.

Registration Authority (RA)

The registration authority interacts with the subscribers. It compiles and verifies the data contained in the certification and/or renewal requests.

The basic data structure for exchanges between KeyOne CA and the authorized RAs is a KeyOne batch. The RA generates a batch with certification/renewal requests that is processed in KeyOne CA in which, in the case of certification requests, a response batch is generated with the certificates created, or, for renewal requests, the change of status of the certificates is confirmed.

The RA can contact an administrator (one with privileges for processing KeyOne batches) and request the processing of the batch in the user interface or directly send the batch to the online batch processing service for an immediate response

1.3.3.2 External Entities

Hardware Security Module (HSM)

To be able to provide its services, KeyOne CA must hold a set of keys, known as the service keys, along with their corresponding certificates. Both these service keys (CRL and certificate signature keys) and the infrastructure keys (batch signing, database integrity, sensitive data encryption, protection of online communications with external systems) are held in the HSM. The HSM performs all the cryptographic operations performed with the service and infrastructure keys.

To protect these keys, the system operator role is defined. Normally, M of a total of N operators are required for enabling access to these keys. Accessing these keys is required for starting the services in the KeyOne CA user interface.

In security terms, service keys are high-risk keys, and should therefore be stored in a FIPS 140-2 level 3 certified HSM (this is required if the system is configured for using a CIMC security policy).

Database Server

The database server is the relational database where KeyOne stores its configuration and the production data.

KeyOne takes advantage of the transactional properties of the databases to assure the atomicity of the changes made to the data when a function is processed.

Cryptographic Device for Authenticating in the User Interface

All KeyOne administrators require a key pair with its associated certificate for authenticating when accessing KeyOne applications. As these are lower risk keys than the service keys, they can be stored on cryptographic tokens (although an HSM can be used).

Administrators must authenticate with their cryptographic devices to access the functions offered by the KeyOne user interface.

The cryptographic device is used for performing cryptographic operations for authenticating administrator.

1.3.3.3 KeyOne System Kernel

KeyOne Kernel System is used by all Keyone products, and specifically by the KeyOne CA product.

1.3.3.3.1 The Application Server

The KeyOne system provides an execution environment or KeyOne application server. This environment supports running and monitoring the services of the KeyOne applications.

1.3.3.3.2 A Shared Security System

Via the KeyOne system, the applications share security mechanisms including:

- User management and authentication.
- Management of privileges for KeyOne applications.
- Management of entity private keys.
- Integrity of records and configuration.

1.3.3.3.3 Management of Private Keys

The private keys used by KeyOne CA are securely managed thanks to mechanisms including the:

- Generation and custody of keys using cryptographic devices (Hardware Cryptographic Modules)[2], online and locally.
- Automatic management of the life-cycle. The coexistence of successively-renewed keys is guaranteed so certificates issued with expired keys can be revoked.

1.3.3.3.4 The Graphical Administration Console

Using the KeyOne Console administration console it is possible to graphically manage aspects of a KeyOne system, such as the:

- Users and roles.
- Access to external resources (e.g., data repositories, cryptographic hardware).

[2] *KeyOne products implement HSM access through appropiate protocols.*

- Installation in multiple machines.
- Recording of events (logs).

1.3.3.3.5 The Security System

The Security System of KeyOne provides of the following security properties:

- EAL4+ Common Criteria (CC) certified production process ([CC_31_Part1], ([CC_31_Part2] and ([CC_31_Part3]).
- Emergency logs.
- An operation mode that forces a configuration as per the NSA/NIST CIMC [CIMC] operating requirements.

The security system (common to all KeyOne products) includes data protection and access-control systems.

1.3.3.3.5.1 The Access-Control System

The access control checks that system users have the required authorization for running the functions accessible via the graphical interface or remote systems.

The KeyOne access system supports operation in [CIMC] or [CWA-14167-1] strict modes; however the evaluated configuration requires [CIMC] operation mode. This means the system can operate as per the recommendations of Directive 1999/93/EC ([EUROPEAN_DIRECTIVE]) and ETSI TS 101 862 for Qualified Certificates ([TS101862]).

1.3.3.3.5.1.1 User Roles

The KeyOne access-control system is based entirely on roles:

- Each KeyOne application defines its actions and the privileges required to execute them. These privileges are grouped in roles defined by the security policy. KeyOne supports defining role incompatibilities.
- Roles are assigned to user groups; this is how users are granted the privileges required to run a certain action.

Although KeyOne applications support adding and customizing roles, the default configuration includes the following roles:

- Administrator

  Administrators have the general responsibility of managing the application. They implement the KeyOne CA security practices and policies. This means that the administrators are responsible for establishing and maintaining KeyOne CA's security configuration.

- Registration Officer (Officer)

  The registration officer role has privileges for processing and sending certificate requests to the system for certificate issue and renewal. Obviously, the registration officer must validate the identity and legitimacy of the data provided by the subscriber.

- Auditor

  Auditors are authorized to browse the production data, audit the log records generated by KeyOne CA and access the application's configuration. Auditors can access any data stored by the application, with the exception of sensitive data such as passwords and private keys.

- System Operator

  The system operator can be seen more as a resource than as a true role as it is only required when the application is started, online or via the user interface. Therefore, it is only required for performing this operation, which requires accessing the service keys of the CA (accessing the HSM). Normally, M operators are defined and only N are required for starting applications.

1.3.3.3.5.2 Data Protection System

The TOE incorporates a proprietary data integrity mechanism called i3D. This mechanism guarantees:

- The integrity of the configuration of the system and the applications.
- The integrity and authenticity of the records. Each data record is signed with a HMAC that has a symmetric key. This guarantees that the entry was generated by the authorized device.
- Detection of loss of records and the unauthorized insertion of records.
- Record historical log. Every change to a record is added to the historical log, guaranteeing the integrity of the life-cycle of the records.

i3D verifies the integrity of the values (past and present) that are stored in the database (e.g., production data, KeyOne registration).

1.3.3.3.5.3 Security Policy

In KeyOne system, security settings are loaded from a security policy that is selected during system startup. The security policy determines the system behaviour regarding the security features of it (the selected policy will determine for example the privilege configuration in KeyOne applications).

The security policies supported by KeyOne system can be the following: Basic, CIMC and CWA 14167-1. In order to guarantee the security conditions and the functional requirements included in this Security Target it is necessary to fix the CIMC security policy.

1.3.4 Conformance Claims

The present Security Target conforms to the following assurance and functional requirements:

- Functional Requirements of the "Part 2: Security functional components" of the Common Criteria Standard. September 2012, Version 3.1, Revision 4.

- Functional Requirements of the Part 2 extended of the Common Criteria Standard. September 2012, Version 3.1, Revision 4.
- Assurance Requirements of the "Part 3: Security assurance components". September 2012, Version 3.1, Revision 4, for the **EAL4 Common Criteria certification level, augmented with ALC_FLR.2**.

This security target claims a demonstrable conformance with **"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).

1.3.4.1 Conformance Rationale

The TOE type described in the 1.2 TOE Overview and 1.3 TOE Description sections of this document is consistent with the TOE type described in the [CIMC] Protection Profile. As indicated in the [CIMC] document, this TOE type basically consists of a Public Key Infrastructure (PKI) that creates and manages public key certificates to facilitate the use of public key cryptography. To achieve this goal, this PKI must perform two basic tasks: 1) generation and distribution of public key certificates to bind public keys to other information after validation of the accuracy of the binding; and 2) maintenance and distribution certificate status information for unexpired certificates.

Because the security problem definition of this Security Target document is exactly the same as the security problem definition of the [CIMC] Protection Profile, it is consistent with the statement of the security problem definition of this Protection Profile. All the assumptions, threats and security policies included in this Security Target are the same as the assumptions, threats and security policies included in the [CIMC] PP.

Likewise, the security objectives of the [CIMC] Protection Profile have been represented exactly in this Security Target document.

The security requirements included in the CIMC Protection Profile have been reproduced in this document. Some of these requirements have been refined and adapted to conform to Version 3.1 of the Common Criteria standard. In all cases, the essence of the Protection Profile´s original requirements has been preserved.

All Security Functional Requirements are equivalent to the [CIMC] Protection Profile except the following, which implement more restrictive functionality than required by this Protection Profile: FAU_GEN.1.1, FMT_MOF.1.1, FDP_ACF.1.2, FPT_CIMC_TSP.1.3, FDP_ACF_CIMC.2.2, FDP_ACF_CIMC.3.1, FCS_CKM_CIMC .5.1. These requirements include Application Notes that justify why the implementation of the TOE on these requirements is stricter from a security point of view than required by the PP.

1.3.5 Legal, Business and Technical Agreements

KeyOne solutions comply with the industry-recognized PKI standards, guaranteeing interoperability with the applications of the main suppliers.

1.3.5.1 Legal Regulations

The KeyOne product family implements a PKI for generating, managing and validating digital certificates and electronic time-stamps. These products facilitate compliance with the European regulatory framework on electronic signatures and the resulting transpossitions in member states, regarding to the certification services that offer these products.

KeyOne CA is a digital certificate generation product. With KeyOne, you can generate the range of certificate types defined in the EU directive ([EUROPEAN_DIRECTIVE]): from the most basic certificates to those offering the maximum legal guarantees (qualified certificates).

KeyOne CA allows generating qualified certificates in accordance with the requirements specified in Annex I "Requirements for qualified certificates" of the EU directive and also enables a certification service provider to comply with the requirements of Annex II "Requirements for certification-service-providers issuing qualified certificates" for public key infrastructure (PKI) products.

KeyOne CA was designed in accordance with the requirements specified in CWA 14167-1 "Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures – Part 1: System Security Requirements" ([CWA-14167-1]). CWA 14167-1 is recognized by the European Commission for electronic signature products that member states consider are in compliance with and that set out to comply with Annex II of Directive 1999/93/EC.

1.3.5.2 Business needs: CEN, ETSI and AIPA/CICA (WebTrust) Regulations

Trusted third parties (TTPs) and certification service providers (CSPs) assure the electronic identification of people and services through the use of digital certificates and electronic signatures. These entities provide a trust framework for the different relationships that come into play in eCommerce and in general in business processes.

Safelayer's products support compliance with the internationally recognized regulations on the policies and requirements that these trust parties can apply.

These regulations and standards include:

- AIPA/CICA – WebTrust Program for Certification Authorities ([WEBTRUST]).
- CWA 14167-1: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements.
- ETSI TS 101 456: Policy Requirements for Certification Authorities issuing qualified certificates ([ETSI_TS_101_456]).
- ETSI TS 102 042: Policy Requirements for Certification Authorities issuing public key certificates ([ETSI_TS_102_042]).
- ETSI TS 102 023: Policy Requirements for time-stamping authorities ([ETSI_TS_102_023]).

1.3.5.3 Business needs: ICAO and EAC ePassport

The ICAO's (International Civil Aviation Organization) ([ICAO_9303]) document 9303 specifies the technical framework for the electronic passport (ePassport or Machine Readable Travel Document, MRTD). The technical specifications in this document were standardized by the ISO/IEC and included in ISO/IEC 7501.

The passport standards cover two generations of electronic passports: first-generation passports (BAC, Basic Access Control) and second-generation passports (EAC, Extended Access Control).

Safelayer offers a complete set of the PKI components required for deploying citizen identification and first- and second-generation ICAO BAC and EAC ePassport solutions. It also provides the technology necessary for managing the National Public Key Directory (N-PKD) and implementing the Single Point of Contact (SPOC) service, a critical component in the security infrastructure required for deploying the new electronic passports in Europe and other countries that facilitates the interconnection of the national PKIs. Safelayer, therefore, has a complete software solution for the PKI of the ePassport, both for the first phase, ICAO/BAC, and the second, EAC.

The KeyOne products for the ICAO/BAC ePassport include the following components:

- CSCA (Country Signing Certification Authority): Manages the digital certificates of the national Document Signers (DSs) and the publication of the ICAO PKD.
- DS (Document Signer): Signs the digitized data stored on the ePassport's chip.

The KeyOne products for EAC ePassport include the following components:

- CVCA (Country Verifying Certification Authority): Issues the CV digital certificates to the Document Verifiers (DV).
- CVRA–SPOC (Country Verifying Registration Authority–Single Point of Contact): Web service interface for the automatic operations (DV certification) and the notifications (suspension of the CVCA service, compromised DV keys, etc.).
- DV (Document Verifier): Acts as a subordinate CA that issues the CV digital certificates to the national inspection systems (IS).
- N-PKD (National Public Key Directory): Manages the ICAO Public Key Directory.

1.3.5.4 Business needs: CIMC Protection Profile

The "Certificate Issuing and Management Components Family of Protection Profiles" protection profile of the US's National Security Agency (NSA) defines requirements for components that issue, revoke and manage public key certificates, such as X.509 public key certificates. These requirements have also been Common Criteria assessed and certified, which endorses the solution's validity.

Safelayer's KeyOne product was designed to comply with the requirements of security level 3 of this protection profile: Certificate Issuing and Management Components (CIMCs) Security Level 3 Protection Profile.

1.3.5.5 Technical Standards

In addition to the requirements and recommendations described in the previous sections, KeyOne complies with all the technical standards for PKI products, of which the following are the most important:

- ITU-T Recommendation X.509 | ISO/IEC 9594-8: "Information Technology – Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks" ([ITU-T X.509v3]).

- RFC 3280, RFC 5280: Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List (CRL) ([RFC3280], [RFC5280]).
- RFC 3039, RFC 3739: Internet X.509 Public Key Infrastructure. Qualified Certificates Profile ([RFC3039], [RFC3739]).
- ETSI TS 101 862: Qualified Certificate Profile ([ETSI_TS_101_862]).
- ETSI TS 102 280: X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons ([ETSI_TS_102_280]).
- RFC 5652: Cryptographic Message Syntax (CMS) ([RFC5652]).
- RFC 4210: Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP) ([RFC4210]).
- RFC 4511: Lightweight Directory Access Protocol (LDAP): The Protocol ([RFC4511]).
- PKCS #1 RSA Cryptography Standard, PKCS #7 Cryptographic Message Syntax Standard, PKCS #10 Certification Request Syntax Standard, PKCS #11 Cryptographic Token Interface Standard, y PKCS #12 Personal Information Exchange Syntax Standard([PKCS#1]).

CHAPTER 2

2 Security Problem Definition

This section includes the following:

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

This information provides the basis for the Security Objectives specified in chapter 3 Security Objectives, and for the Security Requirements for the TOE specified in chapter 4 Security Requirements.

2.1 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).

Assumption Name	Description
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.
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.
Connectivity	
A.Operating System	The operating system has been selected to provide the functions required by this Security Target to counter the perceived threats, as identified in this Security Target.
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.

Table 2-3. Secure Usage Assumptions

2.2 Threats

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

Threat Name	Description
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.
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.
Cryptography	
T.Disclosure of private and secret keys	A private or secret key is improperly disclosed.
T.Modification of private/secret keys	A hacker modifies a secret/private key.
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.
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.

Table 2-4. Threats

2.3 Organizational Security Policies

Policy Name	Description
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.

Table 2-5. Organizational Security Policies

CHAPTER 3

3 Security Objectives

This chapter 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 organizational security policies and assumptions.

3.1 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.

Security Objective for the TOE	Description
Authorized Users	
O.Certificates	The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.
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.
Cryptography	
O.Non-repudiation	Prevent user from avoiding accountability for sending a message by providing evidence that the user sent the message.
External Attacks	
O.Control unknown source	Control (e.g., reroute or discard) communication traffic from an unknown

communication traffic	source to prevent potential damage.

Table 3-1. Security Objectives for the TOE

3.2 Security Objectives for the Environment

This section specifies the security objectives for the environment.

3.2.1 Non-IT security Objectives for the Environment

Non-IT security objective for the environment	Description
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 environment.
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 environment. 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.

Table 3-2. Non-IT Security Objectives for the Environment

3.2.2 IT Security Objectives for the Environment

IT security objective for the environment	Description
O.Cryptographic functions	The IT environment 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.

Table 3-3. IT Security Objectives for the Environment

3.3 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.

Security objective for both the TOE and the Environment	Description
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 for downloads
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.

Table 3-4. Security Objectives for both the TOE and the Environment

CHAPTER 4

4 Security Requirements

4.1 TOE Security Requirements

4.1.1 TOE Security Functional Requirements

This section specifies requirements that are applicable to TOE functionality, such as key management, certificate registration, and product configuration and management functions.

Table 4-1. Functional Requirements for the TOE lists all the functional security requirements for the TOE that are included in this Security Target. The requirements have been extracted from the [CIMC] Protection Profile. Some of these requirements have been instantiated by means the use of the operations mechanism offered by the Common Criteria standard. The following table lists all the security functional requirements for the TOE. *CIMC Access Control Policy is specified in section 10.2 "CIMC TOE Access Control Policy" of [CIMC].*

Functional Requirement
FAU_GEN.1.1
FAU_GEN.1.2
FAU_GEN.2.1
FAU_SEL.1.1
FAU_STG.1.1
FAU_STG.1.2
FAU_STG.4.1
FPT_STM.1.1
FMT_MOF.1.1
FDP_ACC.1.1
FDP_ACF.1.1
FDP_ACF.1.2
FDP_ACF.1.3

FDP_ACF.1.4
FDP_ITT.1.1 (FDP_ITT.1 iteration 1)
FDP_ITT.1.1 (FDP_ITT.1 iteration 2)
FDP_UCT.1.1
FPT_ITC.1.1
FPT_ITT.1.1 (FPT_ITT.1.1 iteration 1)
FPT_ITT.1.1 (FPT_ITT.1.1 iteration 2)
FIA_UAU.1.1
FIA_UAU.1.2
FIA_UID.1.1
FIA_UID.1.2
FIA_USB.1.1
FPT_CIMC_TSP.1.1
FPT_CIMC_TSP.1.2
FPT_CIMC_TSP.1.3
FPT_CIMC_TSP.1.4
FDP_ACF_CIMC.2.1
FDP_ACF_CIMC.2.2
FDP_ACF_CIMC.3.1
FDP_SDI_CIMC.3.1
FDP_SDI_CIMC.3.2
FDP_ETC_CIMC.5.1
FDP_CIMC_BKP.1.1
FDP_CIMC_BKP.1.2
FDP_CIMC_BKP.1.3
FDP_CIMC_BKP.1.4
FDP_CIMC_BKP.2.1
FDP_CIMC_BKP.2.2
FDP_CIMC_CSE.1.1
FDP_CIMC_CER.1.1
FDP_CIMC_CER.1.2
FDP_CIMC_CER.1.3
FDP_CIMC_CER.1.4

FDP_CIMC_CRL.1.1
FDP_CIMC_OCSP.1.1
FCO_NRO_CIMC.3.1
FCO_NRO_CIMC.3.2
FCO_NRO_CIMC.3.3
FCO_NRO_CIMC.4.1
FCO_NRO_CIMC.4.2
FMT_MTD_CIMC.4.1
FMT_MTD_CIMC.5.1
FMT_MTD_CIMC.7.1
FMT_MOF_CIMC.3.1
FMT_MOF_CIMC.3.2
FMT_MOF_CIMC.3.3
FMT_MOF_CIMC.3.4
FMT_MOF_CIMC.5.1
FMT_MOF_CIMC.5.2
FMT_MOF_CIMC.5.3
FMT_MOF_CIMC.6.1
FMT_MOF_CIMC.6.2
FMT_MOF_CIMC.6.3
FCS_CKM_CIMC.5.1
FMT_SMF.1.1
FMT_MTD.1.1 (FMT_MTD.1 iteration 1)
FMT_MTD.1.1 (FMT_MTD.1 iteration 2)
FMT_SMR.1.1
FMT_MSA.1.1 (FMT_MSA.1 iteration 1)
FMT_MSA.1.1 (FMT_MSA.1 iteration 2)
FMT_MSA.3.1
FMT_MSA.3.2
FIA_ATD.1.1

Table 4-1. Functional Requirements for the TOE

4.1.1.1 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.

4.1.1.1.1 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.

4.1.1.1.1.1 FAU_GEN.1 Audit Data Generation

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 [selection, choose one of: *minimum*] level of audit; and

c) [assignment:

- *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 changes to the trusted public keys, including additions and deletions.*
- *The manual entry of secret keys used for authentication.*
- *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[3].*
- *All changes to the certificate revocation list profile.*
- *All changes to the OCSP profile.]*

Application Note: TOE incompatibilities between roles cannot be modified because this would not comply with the CIMC Protection Profile. Consequently, the corresponding event regarding the modification of incompatibilities is not recorded because the event does not occur.

Application Note: The TOE does not retain certificate subject private keys in the TOE. Therefore, it is not possible to access these keys for recovery or other purposes.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST [*the following information:*

[3] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

- *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[4] 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 does not include plaintext private or secret keys or other critical security parameters.

4.1.1.1.1.2 FAU_GEN.2 User identity Association

The TSF shall associate auditable events to individual user identities.

FAU_GEN.2.1

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

4.1.1.1.2 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.

[4] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

4.1.1.1.2.1 FAU_SEL.1 Selective Audit

Selective Audit, requires the ability to include or exclude events from the set of audited events based upon specific attributes.

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*]

4.1.1.1.3 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.

4.1.1.1.3.1 FAU_STG.1 Protected audit trail storage

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 [selection: *detect*] unauthorized modifications to the audit records in the audit trail.

4.1.1.1.3.2 FAU_STG.4 Prevention of audit data loss

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

FAU_STG.4.1

The TSF shall [selection: *prevent audited events, except those taken by the Auditor*] and [assignment: *stores the audited events in a emergency audit repository*] if the audit trail is full.

4.1.1.2 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).

4.1.1.2.1 FPT_STM – Time stamps

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

4.1.1.2.1.1 FPT_STM.1 Reliable time stamps

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.

4.1.1.3 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.

4.1.1.3.1 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.

4.1.1.3.1.1 FMT_MOF.1 Management of security functions behaviour

This component allows the authorized users (roles) to manage the behaviour 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 [selection: *modify the behaviour of*] the functions [assignment: *list of functions listed in the table below*] to [assignment: *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 TSF private keys shall require the authorization of at least two Administrators or*

		one Administrator and one Officer, Auditor or Operator. In this case, the TOE does not allow the export of private keys, and therefore, this restriction about roles it is not applicable.
Certificate Status Change Approval		*Only Officers shall approve the revocation of a certificate or information about the revocation of a certificate.* *Only Officers shall approve the placing of a certificate on hold or information about the hold status of a certificate.*
TSF 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.3 Extended certificate profile management	The capability to modify the certificate profile shall be restricted to Administrators.
Revocation Profile[5] Management		The capability to modify the revocation profile shall be restricted to Administrators.
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 4-2. Authorized Roles for Management of Security Functions Behavior

Application Note: The requirements for private keys export are not applicable because the TOE does not allow the export of this type of keys.

Application Note: Regarding the configuration of the frequency of the periodic signing process, the TOE´s configuration is fixed at the maximum frequency to assure the most secure frequency.

Application Note: Backups are automatically handled in a continuous manner and not on demand. Thus, the system is configured for maximum security because at all times it has all the data required for generating a backup.

[5] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

4.1.1.3.2 FMT_MTD – Management of TSF data

This family allows authorised users (roles) control over the management of TSF data. Examples of TSF data include audit information, clock and other TSF configuration parameters.

4.1.1.3.2.1 FMT_MTD.1 Management of TSF data (iteration 1)

This component allows authorised users to manage TSF data.

FMT_MTD.1.1

The TSF shall restrict the ability to [selection: *query*] the [assignment: *any TSF data*] to [assignment: *auditors*].

4.1.1.3.2.2 FMT_MTD.1 Management of TSF data (iteration 2)

This component allows authorised users to manage TSF data.

FMT_MTD.1.1

The TSF shall restrict the ability to [selection: *change_default, modify, delete and clear]* the [assignment: *any TSF data, except: workflows, logs, issued certificates and CRLs, and processed batches*] to [assignment: *administrators*].

4.1.1.3.3 FMT_SMR – Security Management Roles

This family is intended to control the assignment of different roles to users. The capabilities of these roles with respect to security management are described in the other families in this class.

4.1.1.3.3.1 FMT_SMR.1 Security Roles

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

FMT_SMR.1.1

The TSF shall maintain the roles [assignment: *administrator, registration officer, auditor*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

4.1.1.3.4 FMT_SMF – Specification of Management Functions

This family allows the specification of the management functions to be provided by the TOE.

4.1.1.3.4.1 FMT_SMF.1 Specification of Management Functions

This component requires that the TSF provide specific management functions.

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: [assignment: *Managing of Role/group membership, managing of certification profiles, managing of CRL profiles, managing trust repository for certificate authentication*].

4.1.1.3.5 FMT_MSA – Management of security attributes

This family allows authorised users control over the management of security attributes. This management might include capabilities for viewing and modifying of security attributes.

4.1.1.3.5.1 FMT_MSA.1 Management of security attributes (iteration 1)

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

FMT_MSA.1.1

The TSF shall enforce the [assignment: *CIMC Access Control Policy*] to restrict the ability to [selection: *query*] the security attributes [assignment: *membership of groups, users identity and users authentication certificates*] to [assignment: *auditors and administrators*].

4.1.1.3.5.2 FMT_MSA.1 Management of security attributes (iteration 2)

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

FMT_MSA.1.1

The TSF shall enforce the [assignment: *CIMC Access Control Policy*] to restrict the ability to [selection: *change_default, modify, delete*] the security attributes [assignment: *membership of groups, users identity and users authentication certificates*] to [assignment: *administrators*].

4.1.1.3.5.3 FMT_MSA.3 Static attribute initialisation

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

FMT_MSA.3.1

The TSF shall enforce the [assignment: *CIMC Access Control Policy*] to provide [selection, choose one of: *restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [assignment: *administrator*] to specify alternative initial values to override the default values when an object or information is created.

4.1.1.4 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.

4.1.1.4.1 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.”

4.1.1.4.1.1 FDP_ACC.1 Subset access control

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 [assignment: *CIMC Access Control Policy*] on [assignment: *Subjects: all users of the application successfully identified and authenticated; Objects: configuration data, system and administration data, user data, and operations and function code; Operations: read, write, execute*].

4.1.1.4.2 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.

4.1.1.4.2.1 FDP_ACF.1 Security attribute based access control

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 [assignment: *CIMC Access Control Policy*] to objects based on the following: [assignment: *Subjects: all users of the application successfully identified and authenticated; Objects: configuration data, system and administration data, user data, and operations and function code; SFP-relevant security attributes: identity of the subject requesting access, role or roles the subject is authorized to* *assume, type of access requested, content of the access requests, and possession of a secret or private key if required.]*

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	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	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 Storage	The capability to request the decryption of certificate subject private key 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.
Private and Secret Key Destruction	*The capability to zeroize TSF 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 4-3. Access Controls

Application Note: Because the TOE uses a FIPS 140-2 level 3 validated cryptographic module, component private keys and TSF secret keys cannot be loaded into this cryptographic device.

Application Note: The TOE does not maintain plaintext secret and private keys.

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*].

4.1.1.4.3 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.

4.1.1.4.3.1 FDP_ITT.1 Basic internal transfer protection (iteration 1)

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

FDP_ITT.1.1

The TSF shall enforce the [assignment: *CIMC TOE Access Control Policy*] to prevent the [selection: *modification*] of user data when it is transmitted between physically-separated parts of the TOE.

4.1.1.4.3.2 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 TSF shall enforce the [assignment: *CIMC TOE Access Control*] to prevent the [selection*: disclosure*] of user data when it is transmitted between physically-separated parts of the TOE.

4.1.1.4.4 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.

4.1.1.4.4.1 FDP_UCT.1 Basic data exchange confidentiality

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 [assignment: *CIMC TOE Access Control Policy*] to [selection: *transmit*] user data in a manner protected from unauthorised disclosure.

4.1.1.5 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.

4.1.1.5.1 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.

4.1.1.5.1.1 FPT_ITC.1 Inter-TSF confidentiality during transmission

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 another trusted IT product from unauthorised disclosure during transmission.

4.1.1.5.2 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.

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

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 [selection: *modification*] when it is transmitted between separate parts of the TOE.

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

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 [selection: *disclosure*] when it is transmitted between separate parts of the TOE.

4.1.1.6 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.

4.1.1.6.1 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.

4.1.1.6.1.1 FIA_UAU.1 Timing of authentication

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.

4.1.1.6.2 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.

4.1.1.6.2.1 FIA_UID.1 Timing of identification

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.

4.1.1.6.3 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.

4.1.1.6.3.1 FIA_USB.1 User-subject binding

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 following user security attributes with subjects acting on the behalf of that user: [assignment: *full name, user is enabled, user expiration date, user certificate, list of groups that the user belongs to*].

Application Note: Attributes of the FIA_USB.1.1 requirement are contained in the attributes referenced in the FMT_MSA.1.1 requirement. Note that the "user identity" forms part of the "user is enabled" concept used in the FIA_USB.1.1 requirement and that the user expiration date is considered part of the user's identity.

FIA_USB.1.2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *(1) Subsequent to a successful login by a user, the TSF will assign the user's full name to be the User Identity associated with subjects acting on behalf of the user; (2) Subsequent to a successful* *login by a user, the TSF will use the list of groups that the user belongs to, to determine the User Role associated with subjects acting on behalf of the user*].

FIA_USB.1.3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *when a user's session has been initiated, the security attributes associated with a subject acting on behalf of a user can not be changed during the user's session*].

4.1.1.6.4 FIA_ATD – User attribute definition

All authorised users may have a set of security attributes, other than the user's identity, that is used to enforce the SFRs. This family defines the requirements for associating user security attributes with users as needed to support the TSF in making security decisions.

4.1.1.6.4.1 FIA_ATD.1 User Attribute Definition

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

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *identity, rol, authentication certificate*].

4.1.2 TOE Extended Security Functional Requirements

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

4.1.2.1 FPT – Protection of the TSF

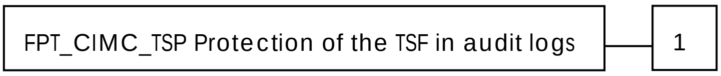
This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may 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.

4.1.2.1.1 FPT_CIMC_TSP - Protection of the TSF in audit logs

Family Behaviour

This family defines requirements that provide integrity mechanisms to the TOE audit logs.

Component levelling



In FPT_CIMC_TSP.1 Audit log signing event, additional protection for stored audit records is required.

Management: FPT_CIMC_TSP.1

There are no management activities foreseen.

Audit: FPT_CIMC_TSP.1

a) Minimal: Digital signature, keyed hash or authentication code shall be included in the audit log.

4.1.2.1.1.1 FPT_CIMC_TSP.1 Audit log signing event

Hierarchical to:

No other components.

Dependencies:

FAU_GEN.1 Audit data generation

FMT_MOF.1 Management of security functions behavior

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Protect stored audit records, by providing additional protection for stored audit records.

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.

Application Note: The TOE's configuration is fixed at the maximum frequency to assure the most secure frequency.

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.

4.1.2.2 FDP – User data protection

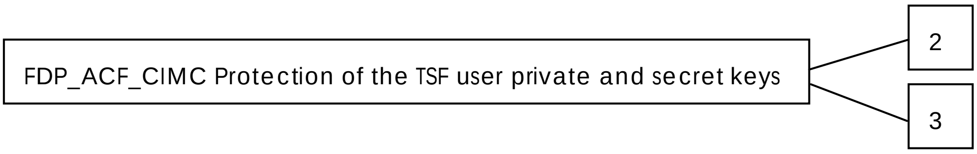
This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data.

4.1.2.2.1 FDP_ACF_CIMC - Protection of the TSF user private and secret keys

Family Behaviour

This family defines requirements that provide confidentiality mechanisms to the TOE user private and secret keys.

Component levelling


FDP_ACF_CIMC Protection of the TSF user private and secret keys
2
3

In FDP_ACF_CIMC.2 additional protection for user private keys is required.

In FDP_ACF_CIMC.3 additional protection for user secreet keys is required.

Management: FDP_ACF_CIMC.2, FDP_ACF_CIMC.3

There are no management activities foreseen.

Audit: FDP_ACF_CIMC.2, FDP_ACF_CIMC.3

There are no auditable events foreseen.

4.1.2.2.1.1 FDP_ACF_CIMC.2 User private key confidentiality protection

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objectives O.Certificates, by ensuring that the certificate is not invalidated by the disclosure of the private key by the TOE, and O.Procedures for preventing malicious code, by ensuring that an untrusted entity can not use a trusted entity's key to sign malicious code.

FDP_ACF_CIMC.2.1

TOE personnel private keys shall be stored in a FIPS 140-1 validated cryptographic module or stored in encrypted form. If TOE 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.

Application Note: TOE personnel private keys are stored in a FIPS 140-1 validated cryptographic module. The TOE does not store certificate subject private keys or user secret keys.

4.1.2.2.1.2 FDP_ACF_CIMC.3 User secret key confidentiality protection

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objectives O.Certificates, by ensuring that 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, and O.Procedures for preventing malicious code, by ensuring that an untrusted entity can not use a trusted entity's key to sign malicious code.

FDP_ACF_CIMC.3.1

User 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.

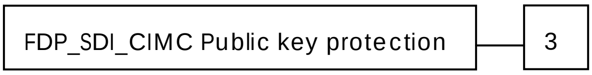
Application Note: TOE personnel private keys are stored in a FIPS 140-1 validated cryptographic module. The TOE does not store certificate subject private keys or user secret keys.

4.1.2.2.2 FDP_SDI_CIMC – Public key protection

Family Behaviour

This family defines requirements that provide integrity mechanisms that protect the TOE public keys against unauthorized modifications.

<u>Component levelling</u>



In FDP_SDI_CIMC.3 Stored public key integrity monitoring and action, protection for unauthorised modification of TOE public keys is required.

Management: FDP_SDI_CIMC.3

There are no management activities foreseen.

Audit: FDP_SDI_CIMC.3

There are no auditable events foreseen.

4.1.2.2.2.1 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 TOE.

Hierarchical to:

No other components.

Dependencies:

No dependencies

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_SDI_CIMC.3.1

Public keys stored within the TOE, 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 [ST assignment: *action to be taken if the verification fails, with the ST rationale showing why this completion is consistent with maintenance of security*].

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*].

4.1.2.2.3 FDP_ETC_CIMC – Extended user private and secret key export

Family Behaviour

This family defines requirements that provide extended user private and secret key export.

Component levelling

FDP_ETC_CIMC Extended user private and secret key export
5

In FDP_ETC_CIMC.5 Extended mechanisms for exportation of the user private and secret keys are required.

Management: FDP_ETC_CIMC.5

There are no management activities foreseen.

Audit: FDP_ETC_CIMC.5

There are no auditable events foreseen.

4.1.2.2.3.1 FDP_ETC_CIMC.5 Extended user private and secret key export

Keys may be exported from cryptographic modules for a variety of reasons, including key backup, replication, and transmission or user private keys generated in the TSF. These security requirements are designed for securing these exportations.

Hierarchical to:

No other components.

Dependencies:

No dependencies

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC. It supports the security objectives O.Data import/export, by covering the requirement that private and secret keys be protected when they are transmitted to and from the TOE.

FDP_ETC_CIMC.5.1

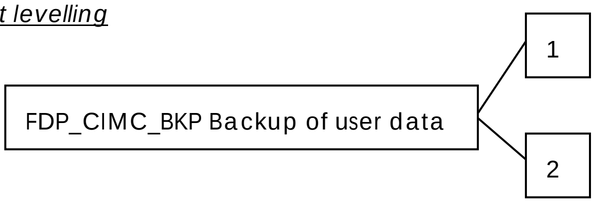
Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

4.1.2.2.4 FDP_CIMC_BKP – Backup and recovery of user data

Family Behaviour

This family defines requirements that provide backup mechanisms of the user data.

Component levelling



In FDP_CIMC_BKP.1 Backup of user data, backup mechanisms are required.

In FDP_CIMC_BKP.2 Backup of user data, recovery mechanisms are required.

Management: FDP_CIMC_BKP.1, FDP_CIMC_BKP.2

There are no management activities foreseen.

Audit: FDP_CIMC_BKP.1, FDP_CIMC_BKP.2

There are no auditable events foreseen.

4.1.2.2.4.1 FDP_CIMC_BKP.1 CIMC backup and recovery

Hierarchical to:

No other components.

Dependencies:

FMT_MOF.1 Management of security functions behavior

Rationale:

This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the CC. It supports the security objectives O.Object and data recovery free from malicious code and O.Preservation/trusted recovery of secure state.

Application Note:

Backups are automatically handled in a continuous manner and not on demand. Thus, the system is configured for maximum security because at all times it has all the data required to generate a backup.

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 TOE 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 TOE transactions has been maintained.

4.1.2.2.4.2 FDP_CIMC_BKP.2 Extended CIMC backup and recovery

Hierarchical to:

No other components.

Dependencies:

FDP_CIMC_BKP.1 CIMC backup and recovery

Rationale:

This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the CC. It supports the security objectives O.Object and data recovery free from malicious code and O.Preservation/trusted recovery of secure state.

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

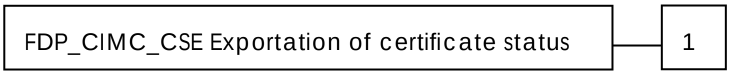
Critical security parameters and other confidential information shall be stored in encrypted form only.

4.1.2.2.5 FDP_CIMC_CSE – Exportation of certificate status

Family Behaviour

This family defines requirements that provide information about the status of the certificates.

Component levelling



In FDP_CIMC_CSE.1 Certificate status export, certificate status exportation mechanisms are required.

Management: FDP_CIMC_CSE.1

There are no management activities foreseen.

Audit: FDP_CIMC_CSE.1

There are no auditable events foreseen.

4.1.2.2.5.1 FDP_CIMC_CSE.1 Certificate status export

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

The TOE must be capable of exporting certificate status information. Any message sent by the TOE 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 [ST assignment: *the X.509 standard for CRLs, the OCSP standard as defined by RFC 2560, other standard (ST shall specify the standard and ST author shall ensure that a description of the format is available), or ST specified format (ST shall include a description of the format)*].

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*].

4.1.2.2.6 FDP_CIMC_CER – CIMC Certificate Generation

Family Behaviour

This family defines requirements for the validation, approval, and signing of public key certificates.

Component levelling

In FDP_CIMC_CER.1 Certificate generation, certificate generation mechanisms are required.

Management: FDP_CIMC_CER.1

There are no management activities foreseen.

Audit: FDP_CIMC_CER.1

a) Minimal: all certificate requests shall be included in the audit log.

4.1.2.2.6.1 FDP_CIMC_CER.1 Certificate Generation

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

The functions in this section address the validation, approval, and signing of public key certificates. X.509 public key certificates issued by the TOE 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 TOE according to the rules of the X.509 standard or validated by the TOE 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:

The data may be approved manually by an Officer.

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

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

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 [ST assignment: *the X.509 standard for public key certificates, other standard (ST shall specify the standard and ST author shall ensure that a description of the format is available), or ST specified format (ST shall include a description of the format)*].

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.

4.1.2.2.7 FDP_CIMC_CRL – CIMC Certificate Revocation List Validation

Family Behaviour

This family defines requirements for the validation and approval of certificate revocation information.

Component levelling

In FDP_CIMC_CRL.1 Certificate revocation list validation, mechanisms of validation and approval of certificate mechanisms are required.

Management: FDP_CIMC_CRL.1

There are no management activities foreseen.

Audit: FDP_CIMC_CRL.1

a) Minimal: all requests to change the status of a certificate must be registered in the audit log.

4.1.2.2.7.1 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 TOE shall be compliant with the X.509 standard. Any fields or extensions to be included in a CRL shall be created by the TOE according to the X.509 standard.

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

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:

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

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

c) 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.

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

e) The thisUpdate field shall indicate the issue date of the CRL.

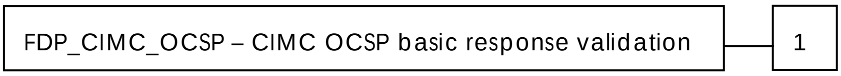
f) The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

4.1.2.2.8 FDP_CIMC_OCSP – CIMC OCSP basic response validation

Family Behaviour

This family defines requirements for the validation and approval of certificate revocation information. OCSP basic responses issued by the TOE shall be compliant with IETF RFC 2560. Any fields or extensions to be included in an OCSP response shall be created by the TOE according to IETF RFC 2560.

Component levelling



FDP_CIMC_OCSP – CIMC OCSP basic response validation

1

In FDP_CIMC_OCSP.1 OCSP basic response validation, mechanisms of validation and approval of certificate mechanisms are required.

Management: FDP_CIMC_OCSP.1

There are no management activities foreseen.

Audit: FDP_CIMC_OCSP.1

There are no auditable events foreseen.

4.1.2.2.8.1 FDP_CIMC_OCSP.1 OCSP basic response validation

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

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC. It supports the security objective O.Certificates, by ensuring that certificate revocation lists and certificate status information are valid.

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.

4.1.2.3 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.

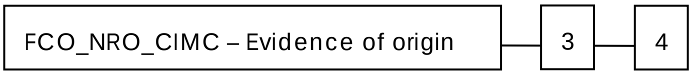
This section also specifies security requirements associated with the export of data from TOEs. The data may be distributed to a device that is outside the boundary of a TOE (either locally or remotely). The remote device or computer may not be directly connected to the TOE. Data export also applies 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.

4.1.2.3.1 FCO_NRO_CIMC – Evidende of origin

Family Behaviour

This family defines requirements that provide proof and verification of origin.

Component levelling



In FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin, requires mechanisms to protect the origin.

In FCO_NRO_CIMC.4 Advanced verification of origin, additional protection of origin is required.

Management: FCO_NRO_CIMC.3, FCO_NRO_CIMC.4

There are no management activities foreseen.

Audit: FCO_NRO_CIMC.3, FCO_NRO_CIMC.4

There are no auditable events foreseen.

4.1.2.3.1.1 FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin

Hierarchical to:

FCO_NRO.2

Dependencies:

FIA_UID.1 Timing of identification

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Non-repudiation and O.Control unknown source communication traffic.

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 [ST assignment: *other attributes*] of the originator of the information, and the security-relevant portions of the information to which the evidence applies.

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.

4.1.2.3.1.2 FCO_NRO_CIMC.4 Advanced verification of origin

Hierarchical to:

FCO_NRO_CIMC.3

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Non-repudiation.

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.

4.1.2.4 FMT – Security management

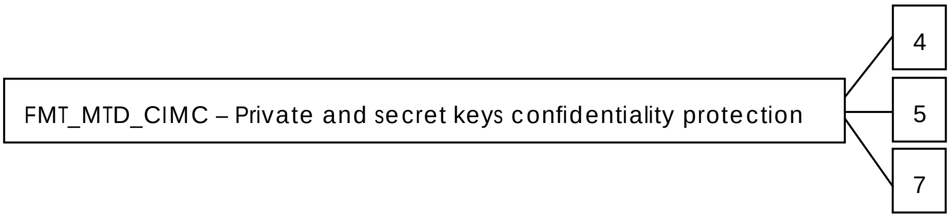
This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

4.1.2.4.1 FMT_MTD_CIMC – Private and secret keys confidentiality protection

Family Behaviour

This family defines requirements that provide confidentiality of TSF private and secret keys.

Component levelling


FMT_MTD_CIMC – Private and secret keys confidentiality protection
4
5
7

In FMT_MTD_CIMC.4 TSF private key confidentiality protection, confidentiality protection of the TSF private keys is requiered.

In FMT_MTD_CIMC.5 TSF secret key confidentiality protection, confidentiality protection of the TSF secret keys is requiered.

In FMT_MTD_CIMC.7 Extended TSF private and secret key export, additional confidentiality protection of the TSF private and secret keys is requiered.

Management: FMT_MTD_CIMC.4, FMT_MTD_CIMC.5, FMT_MTD_CIMC.7

There are no management activities foreseen.

Audit: FMT_MTD_CIMC.4, FMT_MTD_CIMC.5, FMT_MTD_CIMC.7

There are no auditable events foreseen.

4.1.2.4.1.1 FMT_MTD_CIMC.4 TSF private key confidentiality protection

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements.

FMT_MTD_CIMC.4.1

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

4.1.2.4.1.2 FMT_MTD_CIMC.5 TSF secret key confidentiality protection

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements.

Secret (symmetric) keys may be used for several purposes in the TOE. They may be used to encrypt other secret or private keys when they are stored within or exported from the TOE. 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.

4.1.2.4.1.3 FMT_MTD_CIMC.7 Extended TSF private and secret key export

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements.

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 TOE.

FMT_MTD_CIMC.7.1

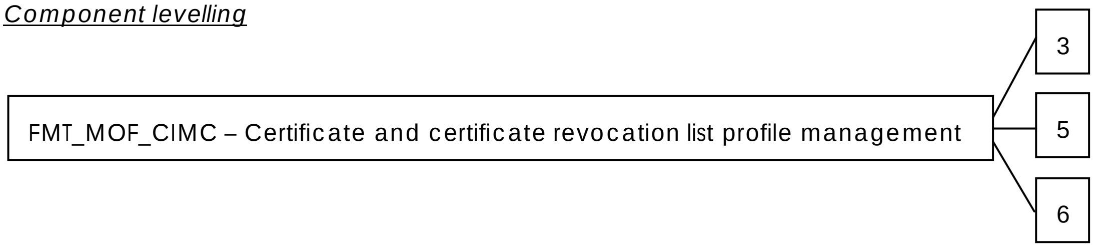
Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

4.1.2.4.2 FMT_MOF_CIMC – Certificate and certificate revocation list profile management

Family Behaviour

This family defines requirements that provide management of certificates and certificate revocation lists profiles.

Component levelling


FMT_MOF_CIMC – Certificate and certificate revocation list profile management
3
5
6

In FMT_MOF_CIMC.3 Extended certificate profile management, requirements of management of certificate profile are required.

In FMT_MOF_CIMC.5 Extended certificate revocation list profile management, management of certificate revocation list profile are required.

Management: FMT_MOF_CIMC.3, FMT_MOF_CIMC.5

There are no management activities foreseen.

Audit: FMT_MOF_CIMC.3, FMT_MOF_CIMC.5

There are no auditable events foreseen.

4.1.2.4.2.1 FMT_MOF_CIMC.3 Extended certificate profile management

Hierarchical to:

No other components.

Dependencies:

FMT_MOF.1 Management of security functions behavior

FMT_SMR.1 Security roles

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Configuration 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.

4.1.2.4.2.2 FMT_MOF_CIMC.5 Extended certificate revocation list profile management

Hierarchical to:

No other components.

Dependencies:

FMT_MOF.1 Management of security functions behavior

FMT_SMR.1 Security roles

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Configuration 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.

4.1.2.4.2.3 FMT_MOF_CIMC.6 OCSP profile management

Hierarchical to:

No other components.

Dependencies:

FMT_MOF.1 Management of security functions behavior

FMT_SMR.1 Security roles

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Configuration 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 TOE 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 TOE 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.

4.1.2.5 FCS – Security management

This class is intended to specify TOE cryptographic functionality.

4.1.2.5.1 FCS_CKM_CIMC – CIMC private and secret key zeroization

Family Behaviour

This family defines requirements for the zeroization/destruction of plaintext private and secret keys stored within the TOE.

Component levelling

FCS_CKM_CIMC – CIMC Private and secret key zeroization
5

In FCS_CKM_CIMC.5 CIMC private and secret key zeroization is required.

Management: FCS_CKM_CIMC.5

There are no management activities foreseen.

Audit: FCS_CKM_CIMC.5

There are no auditable events foreseen.

4.1.2.5.1.1 FCS_CKM_CIMC.5 CIMC private and secret key zeroization

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Rationale:

This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements.

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.

Application Note: The TOE does not maintain plaintext secret and private keys.

4.1.3 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**
Security Target evaluation	ASE_CCL.1, ASE_ECD.1, ASE_INT.1, ASE_OBJ.2, ASE_REQ.2, ASE_SPD.1, ASE_TSS.1
Development	ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3
Guidance Documents	AGD_OPE.1, AGD_PRE.1
Life Cycle Support	ALC_CMC.4, ALC_CMS.4, ALC_DEL.1, ALC_DVS.1, ALC_LCD.1, ALC_TAT.1, ALC_FLR.2
Tests	ATE_COV.2, ATE_FUN.1, ATE_IND.2, ATE_DPT.1
Vulnerability Assessment	AVA_VAN.3

Table 4-4 TOE Security Assurance Requirements

Application note regarding the AGD_PRE.1 and AGD_OPE.1 assurance component: this documentation includes all the necessary information assuring that the software that is downloaded/transferred is inspected prior to being made operational.

The selected set of Security Assurance Requirements is suitable for product certification, since these are the specified by the Common Criteria standard ([CC_31_Part3]) to ensure a level EAL4 +, which is the guaranteed by the TOE.

4.1.3.1 ASE – Security Target evaluation

The objective of this family is to determine the validity of the conformance claim. In addition, this family specifies how Security Targets are to claim conformance with a Protection Profile.

4.1.3.1.1 ASE_CCL.1 Conformance claims

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

4.1.3.1.2 ASE_ECD.1 Extended components definition

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

4.1.3.1.3 ASE_INT.1 ST introduction

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

4.1.3.1.4 ASE_OBJ.2 Security objectives

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

4.1.3.1.5 ASE_REQ.2 Derived security requirements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

4.1.3.1.6 ASE_SPD.1 Security problem definition

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

4.1.3.1.7 ASE_TSS.1 TOE summary specification

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

4.1.3.2 ADV – Development

The requirements of the Development class provide information about the TOE. The Development class encompasses six families of requirements for structuring and representing the TSF at various levels and varying forms of abstraction. These families include: requirements for the description (at the various levels of abstraction) of the design and implementation of the SFRs, requirements for the description of the architecture-oriented features of domain separation, TSF self-protection and non-bypassability of the security functionality, requirements for a security policy model and for correspondence mappings between security policy model and the functional specification, and requirements on the internal structure of the TSF, which covers aspects such as modularity, layering, and minimization of complexity.

4.1.3.2.1 ADV_ARC.1 Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

4.1.3.2.2 ADV_FSP.4 Complete functional specification

ADV_FSP.4.1D The developer shall provide a functional specification.

ADV_FSP.4.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.4.1C The functional specification shall completely represent the TSF.

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

4.1.3.2.3 ADV_IMP.1 Implementation representation of the TSF

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

ADV_IMP.1.1C The implementation representation shall 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 in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

4.1.3.2.4 ADV_TDS.3 Basic modular design

ADV_TDS.3.1D The developer shall provide the design of the TOE.

ADV_TDS.3.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.3.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2C The design shall describe the TSF in terms of modules.

ADV_TDS.3.3C The design shall identify all subsystems of the TSF.

ADV_TDS.3.4C The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7C The design shall describe each SFR-enforcing module in terms of its purpose and relationship with other modules.

ADV_TDS.3.8C The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with other modules and called SFR-related interfaces to other SFR-enforcing modules.

ADV_TDS.3.9C The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10C The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

4.1.3.3 AGD – Guidance documents

The guidance documents class provides the requirements for guidance documentation for all user roles. For the secure preparation and operation of the TOE it is necessary to describe all relevant aspects for the secure handling of the TOE. The class also addresses the possibility of unintended incorrect configuration or handling of the TOE.

4.1.3.3.1 AGD_OPE.1 Operational user guidance

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

4.1.3.4 AGD_PRE.1 Preparative procedures

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

4.1.3.5 ALC - Life-cycle support

Life-cycle support is an aspect of establishing discipline and control in the processes of refinement of the TOE during its development and maintenance. Confidence in the correspondence between the TOE security requirements and the TOE is greater if security analysis and the production of the evidence are done on a regular basis as an integral part of the development and maintenance activities.

4.1.3.5.1 ALC_CMC.4 Production support, acceptance procedures and automation

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

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

ALC_CMC.4.1C The TOE shall be labelled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorised changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

4.1.3.5.2 ALC_CMS.4 Problem tracking CM coverage

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.4.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

4.1.3.5.3 ALC_DEL.1 Delivery procedures

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

4.1.3.5.4 ALC_DVS.1 Identification of security measures

ALC_DVS.1.1D The developer shall produce and provide 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.

4.1.3.5.5 ALC_FLR.2 Flaw reporting procedures

ALC_FLR.2.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

4.1.3.5.6 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.

4.1.3.5.7 ALC_TAT.1 Well-defined development tools

ALC_TAT.1.1D The developer shall provide the documentation identifying each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document and provide the selected implementation-dependent options of each development tool.

ALC_TAT.1.1C Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

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

4.1.3.6 ATE - Tests

The class "Tests" encompasses four families: Coverage, Depth, Independent testing (i.e. functional testing performed by evaluators), and Functional tests. Testing provides assurance that the TSF behaves as described (in the functional specification, TOE design, and implementation representation).

4.1.3.6.1 ATE_COV.2 Analysis of coverage

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 in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

4.1.3.6.2 ATE_DPT.1 Testing: basic design

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

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

4.1.3.6.3 ATE_FUN.1 Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

4.1.3.6.4 ATE_IND.2 Independent testing – sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

4.1.3.7 AVA - Vulnerability Assessment

The AVA Vulnerability assessment class addresses the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE.

4.1.3.7.1 AVA_VAN.3 Focused vulnerability analysis

AVA_VAN.3.1D The developer shall provide the TOE for testing.

AVA_VAN.3.1C The TOE shall be suitable for testing.

4.2 Security requirements for the IT environment

4.2.1 Security Functional Requirements for the IT environment

This section specifies the security functional requirements that are applicable to the IT environment. All these requirements have 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 have 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 4-5. Functional Requirements for the TOE Environment

4.2.1.1 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.

4.2.1.1.1 FAU_GEN – Security Audit Data Generation

4.2.1.1.1.1 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.*]

4.2.1.1.1.2 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.

4.2.1.1.2 FAU_SAR – Security Audit Review

4.2.1.1.2.1 FAU_SAR.1 Audit review

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.

4.2.1.1.2.2 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 4-6. Audit Search Criteria

4.2.1.1.3 FAU_SEL – Security Audit Event Selection

4.2.1.1.3.1 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: a) [selection: *event type*] b) [assignment: *none*].

4.2.1.1.4 FAU_STG – Security Audit Event Storage

4.2.1.1.4.1 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.

4.2.1.1.4.2 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.

4.2.1.2 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).

4.2.1.2.1 FPT_STM – Time stamps

4.2.1.2.1.1 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.

4.2.1.2.2 FPT_SEP – Domain separation

4.2.1.2.2.1 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*].

4.2.1.2.3 FPT_RVM – Reference mediation

4.2.1.2.3.1 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.

4.2.1.2.4 FPT_ITC – Confidentiality of exported TSF data

4.2.1.2.4.1 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.

4.2.1.2.5 FPT_ITT – Internal TOE TSF data transfer

4.2.1.2.5.1 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*].

4.2.1.2.5.2 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*].

4.2.1.2.6 FPT_AMT – Underlying abstract machine test

4.2.1.2.6.1 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*].

4.2.1.3 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.

4.2.1.3.1 FMT_SMR – Security management roles

4.2.1.3.1.1 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*].

4.2.1.3.2 FMT_MOF – Management of functions in TSF

4.2.1.3.2.1 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 4-7. Authorized Roles for Management of Security Functions Behavior

4.2.1.3.3 FMT_MSA – Management of security attributes

4.2.1.3.3.1 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*].

4.2.1.3.3.2 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.

4.2.1.3.3.3 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.

4.2.1.3.4 FMT_MTD – Management of TSF data

4.2.1.3.4.1 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*].

4.2.1.3.5 FMT_SMF – Specification of Management Functions

4.2.1.3.5.1 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*]

4.2.1.4 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.

4.2.1.4.1 FDP_ACC – Access control policy

4.2.1.4.1.1 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*]

4.2.1.4.2 FDP_ACF – Access control functions

4.2.1.4.2.1 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*].

4.2.1.4.3 FDP_ITT – Internal TOE transfer

4.2.1.4.3.1 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*].

4.2.1.4.3.2 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*].

4.2.1.4.4 FDP_UCT – Inter-TSF user data confidentiality transfer protection

4.2.1.4.4.1 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.

4.2.1.5 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.

4.2.1.5.1 FIA_ATD – User attribute definition

4.2.1.5.1.1 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]*].

4.2.1.5.2 FIA_UAU – User Authentication

4.2.1.5.2.1 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.

4.2.1.5.3 FIA_UID – User Identification

4.2.1.5.3.1 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.

4.2.1.5.4 FIA_USB – User-subject binding

4.2.1.5.4.1 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.

4.2.1.5.5 FIA_AFL – Authentication failures

4.2.1.5.5.1 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*].

4.2.1.6 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.

4.2.1.6.1 FTP_TRP – Trusted path

4.2.1.6.1.1 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]*

4.2.1.7 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.

4.2.1.7.1 FCS_CKM – Cryptographic key management

4.2.1.7.1.1 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)]*

4.2.1.7.1.2 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*]

4.2.1.7.2 FCS_COP – Cryptographic operation

4.2.1.7.2.1 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)].*

4.2.2 Proprietary Extended Security Requirements for the IT environment

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.

4.2.2.1.1 FPT_ACC – Access Control

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

4.2.2.1.1.1 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.

4.2.3 Proprietary Extended Security Non-IT Requirements for the environment

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

4.2.3.1 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.

4.2.3.1.1 FPT_ACC – Access Control

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

4.2.3.1.1.1 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.

4.2.4 CIMC Extended Security Functional Requirements

4.2.4.1 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.

4.2.4.1.1.1 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*].

4.2.4.1.1.2 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*].

4.3 Security Rationale

4.3.1 Security Objectives Rationale

This section demonstrates how the Security Objectives trace back to the threats, OSPs and assumptions (security problem definition).

4.3.1.1 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 4-8 Relationships of Security Objectives for the TOE to Threats maps security objectives for the TOE to threats; Table 4-9. Relationship of Security Objectives for the Environment to Threats maps security objectives for the environment to threats; and Table 4-10. Relationship of Security Objectives for Both the TOE and the Environment to Threats maps security objectives for both the TOE and the environment to threats. Table 4-11. Relationship of Security Policies to Security Objectives maps the organizational security policies to security objectives. Table 4-12. Relationship of Assumptions to IT Security Objectives 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 for the TOE	***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 4-8 Relationships of Security Objectives for the TOE to Threats

Security Objective for the Environment	***Threat***
O.Administrators, Operators, Officers and Auditors guidance documentation	T.Administrators, Operators, Officers and Auditors commit errors or hostile actions , T.Social engineering, T.Disclosure of private and secret keys
O.Competent Administrators, Operators,	T.Administrators, Operators, Officers and

Officers and Auditors	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 4-9. Relationship of Security Objectives for the Environment to Threats

Security Objective for both the TOE and the Environment	***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 transfer	T.Message content modification, T.Disclosure of private and secret keys
O.React to detected attacks	T.Hacker gains access
O.Require inspection for downloads	T.Malicious code explotation
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 4-10. 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 4-11. Relationship of Security Policies to Security Objectives

Assumption	***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

Table 4-12. Relationship of Assumptions to IT Security Objectives

4.3.1.2 Security Objectives Sufficiency

The following sections provide information regarding:

Why the identified security objectives provide for effective countermeasures to the threats;

Why the identified security objectives provide complete coverage of each organizational security policy;

Why the identified security objectives uphold each assumption.

4.3.1.2.1 Threats and Objectives Sufficiency

4.3.1.2.1.1 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 ensures that security-relevant software, hardware, and firmware are correctly functioning through features and procedures such as underlying machine testing and integrity checks.

4.3.1.2.1.2 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 ensures 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.

4.3.1.2.1.3 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 IT environment 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 IT environment 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.

4.3.1.2.1.4 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.

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.

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.

4.3.1.2.2 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.

4.3.1.2.3 Assumptions and Objectives Sufficiency

4.3.1.2.3.1 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.

4.3.1.2.3.2 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.

4.3.1.2.3.3 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.

4.3.2 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.

4.3.2.1 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 Table 4-13. Security Functional Requirements Related to Security Objectives, addresses the mapping of security functional requirements to security objectives. The Table 4-14. Security Assurance Requirements Related to Security Objectives, addresses the mapping of security assurance requirements to security objectives.

Functional Requirement	***Objective***
FAU_GEN.1 Audit data generation	O.Individual accountability and audit records
FAU_GEN.2 User identity association	O.Individual accountability and audit records
FAU_SEL.1 Selective Audit	O.Individual accountability and audit records
FCS_CKM_CIMC.5 CIMC private and secret key zeroization	O.Procedures for preventing malicious code, O.React to detected attacks
FMT_MTD.1 Management of TSF data (iteration	O.Individual accountability and audit records

1, iteration 2)	O.Protect stored audit records
FMT_SMF.1 Specification of Management Functions	O.Individual accountability and audit records
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_ETC_CIMC.5 Extended user private and secret key export	O.Data import/export
FAU_STG.1 Protected audit trail storage	O.Protect stored audit records
FAU_STG.4 Prevention of audit data loss	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
FDP_ACC.1 Subset access control	O.Limitation of administrative access
FDP_ACF.1 Security attribute based access control	O.Limitation of administrative access
FMT_MSA.1 Management of security attributes (iteration 1, iteration 2)	O.Limitation of administrative access O.User authorization management O.Maintain user attributes O.Security-relevant configuration management
FMT_MSA.3 Static attribute initialisation	O.Limitation of administrative access O.Maintain user attributes O.Security-relevant configuration management O.User authorization management
FMT_SMR.1 Security Roles	O.Limitation of administrative access
FMT_SMF.1 Specification of Management Functions	O.Limitation of administrative access
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	O.Certificates

validation	
FDP_CIMC_OCSP.1 OCSP basic response validation	O.Certificates
FDP_CIMC_CSE.1 Certificate status export	O.Certificates
FDP_ITT.1 Basic internal transfer protection (iteration 1)	O.Integrity protection of user data and software, O.Protect user and TSF data during internal transfer
FDP_ITT.1 Basic internal transfer protection (iteration 2)	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	O.Data import/export
FIA_UAU.1 Timing of authentication	O.Limitation of administrative access, O.Restrict actions before authentication
FIA_UID.1 Timing of identification	O.Individual accountability and audit records, O.Limitation of administrative access
FIA_USB.1 User-subject binding	O.Maintain user attributes
FIA_ATD.1 User attribute definition	O.Maintain user attributes
FMT_MOF.1 Management of security functions behavior	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_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
FPT_CIMC_TSP.1 Audit log signing event	O.Protect stored audit records
FPT_ITC.1 Inter-TSF confidentiality during transmission	O.Data import/export
FPT_ITT.1 Basic internal TSF data transfer protection (iterations 1-2)	O.Protect user and TSF data during internal transfer
FPT_STM.1 Reliable time stamps	O.Individual accountability and audit records, O.Time stamps
FPT_RVM.1 Non-bypassability of the TSP	O.Operating System

Table 4-13. Security Functional Requirements Related to Security Objectives

Assurance Requirement	***Objective***
ADV_ARC.1 Security architecture description	EAL4,
ADV_FSP.4 Complete functional specification	EAL4, O.Lifecycle security
ADV_IMP.1 Implementation representation of the TSF	EAL4, O.Lifecycle security
ADV_TDS.3 Basic modular design	EAL4, O.Lifecycle security
AGD_OPE.1 Operational user guidance	EAL4, 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,
AGD_PRE.1 Preparative procedures	EAL4, O.Installation
ALC_CMC.4 Production support, acceptance procedures and automation	EAL4, O.Configuration Management
ALC_CMS.4 Problem tracking CM coverage	EAL4, O.Configuration Management
ALC_DEL.1 Delivery procedures	EAL4
ALC_DVS.1 Identification of security measures	EAL4
ALC_LCD.1 Developer defined life-cycle model	EAL4
ALC_TAT.1 Well-defined development tools	EAL4
ATE_COV.2 Analysis of coverage	EAL4
ATE_DPT.1 Testing: basic design	EAL4

ATE_FUN.1 Functional testing	EAL4
ATE_IND.2 Independent testing - sample	EAL4
AVA_VAN.3 Focused vulnerability analysis	EAL4
ALC_FLR.2 Flaw reporting procedures	EAL4+ALC_FLR.2 O.Lifecycle security O.Repair identified security flaws

Table 4-14. Security Assurance Requirements Related to Security Objectives

Although the FPT_RVM.1 component was included in the [CIMC] Protection Profile, because this component has not been maintained in the 3.1 version of the Common Criteria standard, this Security Target does not formally claim the FPT_RVM.1.1 requirement. The objective of this requirement is the non-bypassability for all SFPs in the TSP, and this objective is assured by the ADV_ARC.1 assurance component, included in the scope of this Security Target.

4.3.2.2 Security Requirements Sufficiency

4.3.2.2.1 Security Objectives for the TOE

4.3.2.2.1.1 Authorized users

O.Certificates is provided by **FDP_CIMC_CER.1 (Certificate Generation)** which ensures that certificates are valid, **FDP_CIMC_CRL.1 (Certificate revocation list validation),** which ensures that certificate revocation lists are valid, **FDP_CIMC_CSE.1 (Certificate status export)** which ensure that certificate status information are valid, and **FDP_CIMC_OCSP.1 (OCSP basic response validation)** which ensure that certificates are not revoked. The TOE does not store neither certificate subject private keys nor user secret keys, and therefore, it guarantees the compliance of FDP_ACF_CIMC.2 (User private key confidentiality protection) ensuring that the TOE personnel private keys are stored in FIPS 140-1 validated cryptographic module. In this case, FDP_ACF_CIMC.2 guarantees that the certificate is not invalidated by the disclosure of the private key by the TOE. 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. In this case the TOE does not store neither certificate subject private keys nor user secret keys; TOE personnel private keys are stored in FIPS 140-1 validated cryptographic module.

4.3.2.2.1.2 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.

4.3.2.2.1.3 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.

4.3.2.2.1.4 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.

4.3.2.2.2 Non-IT Security Objectives for the Environment

O.Administrators, Operators, Officers and Auditors guidance documentation is provided by **AGD_OPE.1 (Operational User Guidance)** which ensures that adequate guidance on the secure operation of the TOE is provided to all types of users of the TOE: end-users, persons responsible for maintaining and administering the TOE, and by others (e.g. programmers) using the TOE's external interfaces.

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_OPE.1 (Operational User 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_OPE.1 (Operational User Guidance)** which ensures that adequate guidance on the secure operation of the TOE is provided to all types of users of the TOE: end-users, persons responsible for maintaining and administering the TOE, and by others (e.g. programmers) using the TOE's external interfaces.

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 **AGD_OPE (Operational User Guidance)** and **AGD_PRE (Preparative Procedures)** 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_OPE.1 (Operational 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.4 (Complete Functional Specification), ADV_IMP.1 (Implementation representation of the TSF), ADV_TDS.3 (Basic modular design)**. **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.

4.3.2.2.3 Security Objectives for the TOE and the Environment

O.Configuration Management is provided by **FMT_MOF.1 (Management of security functions behavior)** 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_OPE.1 (Operational User 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 **ALC_CMS.4 (Problem tracking CM coverage)** and **ALC_CMC.4 (Production support, acceptance procedures and**

automation) 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)** and **FPT_ITC.1 (Inter-TSF confidentiality during transmission)** which cover the requirement that data other than private and secret keys be protected when they are transmitted and from the TOE. **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, **FDP_CIMC_BKP.2 (Extended CIMC backup and recovery)** which covers the requirement that modifications to backup data be detected. Since **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)** and **FIA_UID.1 (Timing of identification) (Iteration 1)** cover the requirement that users be identified before performing any security-relevant operations. **FAU_GEN.1 (Audit data generation), FAU_GEN.1 (Audit data generation) (Iteration 1), FAU_SEL.1 (Selective audit) and FAU_SEL.1 (Selective audit) (Iteration 1)** cover the requirement that security-relevant events be audited while **FAU_GEN.2 (User identity association), FAU_GEN.2 (User identity association) (Iteration 1), FPT_STM.1 (Reliable time stamps) and FPT_STM.1 (Reliable time stamps) (Iteration 1)** 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)** that allows authorized users to manage TSF data, and **FMT_SMF.1 (Specification of Management Functions)** that requires that the TSF provide specific management functions. 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)** 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)** which covers 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)**, **FDP_ACF.1 (Security attribute based access control)**, **FIA_UAU.1 (Timing of authentication)**, **FIA_UID.1 (Timing of identification)**, **FMT_MSA.3 (Static attribute initialisation), FMT_MSA.1 (Management of security attributes) FMT_SMR.1 (Security Roles), and FMT_SMF.1 (Specification of Management Functions)**. **FIA_UAU.1 (Timing of**

authentication) and **FIA_UID.1 (Timing of identification)** 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)** and **FDP_ACF.1 (Security attribute based access control)** ensure that Administrators, Operators, Officers, and Auditors can only perform those operations necessary to perform their jobs. **FMT_MSA.1 (Management of security attributes)** allows authorized users (roles) to manage the specified security attributes, and **FMT_MSA.3 (Static attribute initialization)** ensures that the default values of security attributes are appropriately either permissive or restrictive in nature. **FMT_SMR.1 (Security Roles)** specifies the roles with respect to security that the TSF recognizes. **FMT_SMF.1 (Specification of Management Functions)** requires that the TSF provide specific management functions.

O.Maintain user attributes is provided by **FIA_USB.1 (User-subject binding)** which cover the requirement to maintain a set of security attributes associated with individual users and/or subjects acting on users' behalves, by **FIA_ATD.1 (User attribute definition)** that allows user security attributes for each user to be maintained individually, by **FMT_MSA.1 (Management of security attributes)** that allows authorized users (roles) to manage the specified security attributes, and by **FMT_MSA.3 (Static attribute initialization)** ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.

O.Manage behavior of security functions is provided by **FMT_MOF.1 (Management of security functions behavior)** which covers the requirement that authorized users be able to configure, operate, and maintain the security mechanisms.

O.Object and data recovery free from malicious code is provided by **FDP_CIMC_BKP.1 (CIMC backup and recovery), FDP_CIMC_BKP.2 (Extended CIMC backup and recovery)** which cover the requirement to be able to recover to a viable state, and **FPT_TST_CIMC.2 (Software/firmware integrity test)** which covers the requirement that the recovered state is free from malicious code.

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_OPE.1 (Operational 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)** 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)** which covers the requirement that audit records be protected against modification or unauthorized deletion. **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. **FMT_MTD.1 (Management of TSF data)** (iteration 1 and 2) is also needed because allows only authorized users to manage TSF data.

O.Protect user and TSF data during internal transfer is provided by **FDP_ITT.1 (Basic internal transfer protection)** which covers the requirement that user data be protected during internal transfer and **FPT_ITT.1 (Basic internal TSF data transfer protection)** 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_OPE.1 (Operational User Guidance), AGD_PRE.1 (Preparative Procedures)** 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)** 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)** 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_MOF.1 (Management of security functions behavior)** 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_OPE.1 (Operational User Guidance)** which covers the requirement that Administrators be provided with documentation describing the configuration management features of the TOE, 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.Security-relevant configuration management** is also supported by **FMT_MSA.1 (Management of security attributes)** (iteration 1) that allows authorized users to manage the specific security attributes, and by **FMT_MSA.3 (Static attribute initialization)** that ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.

O.Time stamps is provided by **FPT_STM.1 (Reliable time stamps)** which covers the requirement that the time stamps be reliable.

O.User authorization management is provided by **AGD_OPE.1 (Operational User Guidance)** which covers the requirement that Administrators be provided with documentation describing the user authorization management features of the TOE, by **A.Competent Administrators, Operators, Officers and Auditors**, by **A.CPS** which ensure that Administrators are competent and are familiar with the CPS under which the TOE is to be operated, and by **FMT_MSA.1 (Management of security attributes)** that allows authorized users to manage the specific security attributes. **O.User authorization management** is also supported by **FMT_MSA.1 (Management of security attributes)** (iteration 1) that allows authorized users to manage the specific security attributes, and by **FMT_MSA.3 (Static attribute initialization)** that ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.

O.React to detected attacks is provided by **FCS_CKM_CIMC.5 (CIMC private and secret key zeroization)** which covers 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.

4.3.3 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 demonstrated in an analysis of dependencies. Mutual support is shown through consideration of the interactions between and among the SFRs.

4.3.3.1 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.

The Table 4-15. Summary of Security Functional Requirements Dependencies provides 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_SEL.1 Selective Audit	FAU_GEN.1 Audit data generation	Included
	FMT_MTD.1 Management of TSF data	Included
FMT_MTD.1 Management of TSF data	FMT_SMR.1 Security roles	Included
	FMT_SMF.1 Specification of management functions	Included
FMT_SMR.1 Security roles	FIA_UID.1 Timing of identification	Included
FMT_SMF.1 Specification of management functions	None	
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_CIMC.5 CIMC private and secret key zeroization	FCS_CKM.4 Cryptographic key destruction	Not satisfied (the destruction of the keys is done directly by the HSM)
	FDP_ACF.1 Security attribute based access control	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
FMT_MSA.3 Static attribute initialization	FMT_MSA.1 Management of security attributes	Included
	FMT_SMR.1 Security roles	Included
FIA_ATD.1 User attribute definition	None	
FDP_ACF_CIMC.2 User private key confidentiality protection	None	
FDP_ACF_CIMC.3 User secret key confidentiality protection	None	
FMT_MSA.1 Management of security attributes	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_ACC.1 Included
	FMT_SMR.1 Security roles	Included
	FMT_SMF.1 Specification of Management Functions	Included
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_OCSP.1 OCSP basic response validation	None	
FDP_ETC_CIMC.5 Extended user private and secret key export	None	
FDP_CIMC_CSE.1 Certificate status 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	FDP_ACC.1 Included
	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path	Not satisfied. These trusted communication requirements are supported by the environment components.
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
	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
FMT_MOF_CIMC.5 Extended certificate revocation list profile management	FMT_MOF.1 Management of security functions behavior	Included
	FMT_SMR.1 Security roles	Included
FMT_MOF_CIMC.6 OCSP profile management	FMT_MOF.1 Management of security functions behaviour	Included
	FMT_SMR.1 Security roles	Included
FMT_MTD_CIMC.4 TSF private key confidentiality protection	None	
FMT_MTD_CIMC.5 TSF secret key confidentiality protection	None	
FMT_MTD_CIMC.7 Extended TSF private and secret key export	None	Included
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	

FDP_ACC.1 Access Control	FDP_ACF.1 Security attribute based access control	

Table 4-15. Summary of Security Functional Requirements Dependencies

4.3.3.1.1 Justification of Unsupported Dependencies Regarding FTP_ITC.1 or FTP_TRP.1

Component FDP_UCT.1 Basic data exchange confidentiality has a direct dependency on FTP_ITC.1 Inter-TSF trusted channel or FTP_TRP.1 Trusted path that is unmet. This product uses basic encryption to ensure basic data exchange confidentiality. It is unnecessary for this product to require Inter-TSF trusted channel or trusted path at this Security Level.

CHAPTER 5

5 TOE Summary Specification

This section describes how the TOE satisfies all the Security Functional Requirements identified in TOE Security Requirements

TOE Security Functional Requirements, page 31, providing the general technical mechanism that the TOE uses for this purpose.

5.1 Audit Data Management

KeyOne 4.0 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).

5.1.1 Functional requirements satisfied by the TOE

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

- Selective Logs Function. This functionality allows configuring the events to audit. The KeyOne Console application has 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 is able to register.
- Security Audit Data Generation Function. 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:

FAU_GEN.1.1

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, and when any of the KeyOne Applications are started/stopped. It is not possible with KeyOne products to start only the Security Audit Data Generation Services without to start the KeyOne Server or KeyOne Application, and it is not possible to shutdown the Security Audit Data Generation Function without to shutdown the KeyOne Server or KeyOne Application. When the KeyOne server or KeyOne Application starts, an audit record is generated in the i3D Database Logs Table, and when the KeyOne server or KeyOne Application 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. All changes to this environment clock are collected by the KeyOne system, in order to incorporate the new system clock in the new logs registered by the TOE.

   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 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 a Registration Authority (for instance, KeyOne LRA) and KeyOne CA, and the communication between the KeyOne CA and a Validation Authoritt (for instance 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 a Registration Authority 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.

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 the KeyOne applications and the database. These communications imply the creation of a 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 . 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 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. 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).

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 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 TOE context, these events are the ones related to the reception of information from a Registration Authority or from a Validation Authority.

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

The TOE exports security configuration data, as the certification and revocation profile[6] or logs configurations. This information is exported in an XML file, along with a file containing the hash of this information.

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. The TOE allows only import keys to the hardware cryptographic module (keys generated in the same cryptographic module).

g) 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.

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

[6] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

i) 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 CA are registered by the Security Audit Data Generation Function in audit logs.

j) 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 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

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

l) All changes to the revocation profile[7] (Revocation Profile Management). The Security Audit Data Generation Function generates a log entry for each change to the revocation profile.

m) All changes to the certificate revocation list profile (FMT_MOF_CIMC.5). The Security Audit Data Generation Function generates a log entry for each change to the revocation profile[8].

n) All changes to the OCSP profile.

[7] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

[8] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

It is not possible to modity TOE incompatibilities between roles, because if not, do not comply with the CIMC Protection Profile. Consequently, the corresponding event regarding modification of incompatibilities is not recorded because the event does not occur.

Regarding to the security events corresponding to the key recovery, the TOE does not retain certificate subject private keys within the TOE, and therefore it is not possible to access to these keys for recovery or other purposes.

FAU_GEN.1.2

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

- 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.
- A 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. The Security Audit Data Generation Function includes this component in the following operations:
  - Request of asymmetric key pair 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.3).
- For each change to the revocation profile[9]: 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.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.

As explained in section 5.2 Secure Database, the integrity of the Databases is provided through the i3D tools. The TOE provides a set of tools to verify the integrity of all Databases managed by the product. These integrity verification tools are invoked on demand, from a frequency of verification that should be established in each context of use of the TOE.

FAU_GEN.2.1

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.

FAU_SEL.1.1

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 allows changing from this option the events to include/exclude from the showed list.

5.2 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.

[9] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

- 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.

5.2.1 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.

5.2.1.1 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.

5.2.1.2 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.

5.2.1.3 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.

5.2.1.4 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.

5.2.2 Functional requirements satisfied by the TOE

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

- Database Integrity Verification Function. 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. This service allows manage the KeyOne services when the database capacity is full.

- Session Table Management. 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. 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:

FAU_STG.1.1

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

FAU_STG.1.2

By means the Database Integrity Verification Function, the TOE is able to detect modifications to the database records, and therefore it allows detect modifications to the audit records.

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 not stored 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 of a public key not certified, between a KeyOne XRA and KeyOne CA, the integrity of this key is provided by means the signed format that is used for this communication (KeyOne batch) (FDP_SDI_CIMC.3.1), and for the integrity mechanism provided by the SSL/TLS protocol (FDP_SDI_CIMC.3.1). Because the public keys that are not certified are store in the database of the KeyOne system, then these public keys are protected by means the integrity provided by the i3D database.

The Session Table Management Function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at the Starting an i3D session, page 134, 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 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, page 134, 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.

FPT_STM.1.1

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

The TOE implements security functionality to generate a stamp of reliable-time. In this case, this function is performed by the KeyOne i3D technology by ensuring through a stamp (integrity) the reliable-times managed by the TSF. The "reliability" of these times is not a intrinsic property of the TOE, but the property is integrity or "stamping".

That is, the security property of the FPT_STM.1.1 requirement is the stamping of the reliable-times. In this case, the TOE needs reliable-times that are generated by the environment.

FPT_CIMC_TSP.1.1, FPT_CIMC_TSP.1.2, 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. 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 134, 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.

FAU_STG.4.1

This requirement forces to prevent auditable events if the audit trail is full.

In fact the behavior of the TOE is the same in any situation in which the product cannot record logs in the database, regardless of the specific error (database full, connection error with the database, ...).

In this case, if the audit trail is full, the Check Database Capacity Function manages the control of the situation. This function is in charge of t:

- Generating a protected auxiliary log record on disk. In this case, the record indicates that the audit trail is full.

- Rolling-back the actions performed related to the event.

- Stopping the system.

When emergency logs are activated, whenever an attempt is made to start a transaction, the contents of the emergency logs are moved to the database. If that is not possible, an error log is recorded in the emergency error repository.

When a KeyOne application is started, it checks if emergency logs are pending to be moved to the database; if so, an attempt is made to move them to the database. If errors occur in this process, they are recorded as new emergency log records, and only Auditor role users can access to the application. The Auditor role can examine the contents of these emergency logs using a log viewing utility. Access control limits the use of this visualization utility and restricts its use to the Auditor role. Furthermore, use of this utility is also registered in the emergency logs.

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. 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 134, 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 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.

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).

FDP_CIMC_BKP.2.1

This requirement forces to protect the backup data 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 information stored in the KeyOne Databases. This information is protected by means the KeyOne i3D mechanism and making use of the I3D session and I3D historic security functions.

The audit log-signing event is generated by the Session Table Management Function This function generates the asymmetrical digital signature from the user's digital signature certificate. As it is explained at Starting an i3D session section, page 134, 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.

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 against modification through the use of digital signatures.

5.3 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.

5.3.1 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.

5.3.2 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.

5.3.3 Functional requirements satisfied by the TOE

The Access Control Management services are composed of the following security function:

- Access Control Function. 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:

FMT_MOF.1.1

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/Authorised role
Security Audit		The capability to configure the audit parameters shall be restricted to Administrators.
Recovery		The capability to configure the recovery parameters shall be restricted to Administrators. The capability to initiate the recovery function shall be restricted to anyone with access to the database.
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.
Certificate Status Change Approval		Only Officers shall approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall approve the placing of a certificate on hold or information about the hold status of a certificate.
TSF 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.3 Extended certificate profile management	The capability to modify the certificate profile shall be restricted to Administrators.
Revocation Profile[10] Management		The capability to modify the revocation profile[11] shall be restricted to Administrators.
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 Behaviour

The relationships between the operations and roles are established in the security policy. By modifying the KeyOne security policy is possible to set the privileges to the appropriate role that will be able to execute the related operation.

FDP_ACC.1.1, FMT_MTD.1.1 (iteration 1), FMT_MTD.1.1 (iteration 2), FMT_SMR.1.1

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

[10] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

[11] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

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.

FDP_ACF.1.1

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).

FDP_ACF.1.2

The KeyOne Access Control can be configured in order to conform the following rules specified in the table below:

Section/function	Component	Function/Authorised 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		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.
Trusted Public Key Entry, Deletion, and Storage		The capability to change (add, revise, delete) the trusted public keys shall be restricted to Administrators.
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

role that can access to these operations can be configured in the security policy.

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).

FMT_MOF_CIMC.3.2, FMT_MOF_CIMC.3.3, FMT_MOF_CIMC.3.4

The KeyOne system can be configured in order to determine that a specific role could assign the acceptable values for the certificate fields and extensions. This configuration can only be performed by the administrator role. An specific 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 specific privilege.

FMT_MOF_CIMC.5.2, 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. This configuration can only be performed by the administrator role. The specific 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 specific privilege.

FDP_ACF.1.3, FDP_ACF.1.4

This requirement does not imply the use of any access control mechanism, and therefore there are not any TOE security function related to it.

5.4 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 156.

5.4.1 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 a smart card initialised by Safelayer (it is delivered with the CD) or issued by another CA.

b) The Security Officer does not have a 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.

5.4.2 Special groups of users

There are the following groups of users that are managed in a special way:

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.

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 carry 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.

5.4.3 Authentication modes

The following user authentication modes are defined:

- Certificate (smart card).
- Username and password (only allowed for Administrators in an emergency situation in which authentication is not possible from cards).

- 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.

5.4.4 Functional requirements satisfied by security functions

The Identification and Authentication services are composed of the User Identification and Authentication security function. 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:

FIA_UAU.1.1, FIA_UAU.1.2

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). 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 or authentication through certificate. If a CIMC policy has been configured, then the authentication through certificate is mandatory and it is not possible to use a password in the authentication process.

In this case, the "`card`" value from the `Authentication mode` list is activated. 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) is erroneous, an error message is shown on screen. In this case, the user must re-enter the certificate and associated PIN (if the number of unsuccessful authentication attempts equals or surpasses the maximum number of allowed attempts, the TOE prevents further authentication attempts).

In the login procedure, the user can abort the process before the completion of the authentication phase and before sending the user credencials.

If the Administrator loses the ability to authenticate (for examplebecause of a malfunctioning card reader), an emergency login mechanism is provided to resolve this type of incident. This mechanism consists of a recovery password that only can be used in KeyOne Console. In these situations, Administrators can also access with their personal passwords.

FIA_UID.1.1, FIA_UID.1.2

Depending on how the user has been configured he will be able to carry out the identification through username or through certificate (cryptographic card). 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 or identification through certificate. If a CIMC policy has been configured, then the identification through certificate is mandatory and it is not possible to use a username in the identification process.In this case, the "`card`" value from the `Authentication mode` list is activated.

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 and before sending the user credencials.

In case that Administrator lose the ability to authenticate (for example, by a malfunctioning of a card reader), an emergency login mechanism is provided to solve this type of incident. This mechanism includes a recovery password that only can be used from KeyOne Console.

FIA_USB.1.1, FIA_USB.1.2, FIA_USB.1.3

The User Identification and Authentication Function after identifying and authenticating the user, it associates the appropriate user security attributes (full name, user is enabled, user expiration date, user certificate, list of groups that the user belongs to) 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.

When a CIMC Policy is configured, then the authentication mode is by using a certificate. In this case, 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).

Once the initial association of user security attributes and subject has been established, then the TSF will assign the user's full name with the subject acting on behalf of the user. Furthermore the TSF will use the list of groups that the user belongs to, to determine the user role associated with subjects acting on behalf of the user. When a user's session has been initiated, the security attributes associated with a subject acting on behalf of a user can not be changed during the user's session.

5.5 Management of security parameters and functions

The management of security settings, and the management of security functions, are based on the access control of the system. KeyOne protects access to sensitive parameters and functions through the management of roles implanted in the system. The default roles defined on a KeyOne system configured with a CIMC Security Policy are the following:

Role	*User attributions*
Administrator	Performs administrative sessions and actions that modify security functions and other settings.
Auditor	Audits and inspects operations performed in the application instance configuration (e.g. browse the log registry).
Registration Officer	Verifies and registers data provided by end entities during registration; performs requests for certificate issuance, revocation, suspension or enablement requests.

5.5.1 Functions of KeyOne System

KeyOne system functions can be classified in two main categories:

- Functions that provide PKI services (e.g. generating a certificate).
- Security functions for protecting PKI services provided by applications.

5.5.1.1 Security Functions of the KeyOne System

The security functions of the KeyOne system protect the PKI services provided by the applications. The main security functions are:

- To guarantee the integrity and authenticity of the code that KeyOne applications execute.
- To guarantee the integrity and authenticity of the databases that KeyOne applications use.
- To guarantee the integrity, authenticity and confidentiality of communications between different KeyOne applications.
- To provide access control of actions performed in KeyOne applications.
- To authenticate KeyOne application users.
- To record events and actions performed in KeyOne applications and facilitate its auditing.
- To protect access to cryptographic keys that the PKI services use (e.g. keys for issuing certificates).
- To protect access to cryptographic keys that the security functions use (e.g. obfuscation keys, keys for signing batches, keys for singing i3D sessions).
- To guarantee the integrity, authenticity and confidentiality of the system configuration.

5.5.2 KeyOne Configuration

KeyOne applications access configuration data that needs to be made secure as this data affects the behavior of system functions.

The main configuration data of the KeyOne system comprises:

- Roles existing in the applications.
- Permissions that roles have for application functions.
- Users and user groups existing in the applications.
- Groups to which the users of the applications belong.
- Roles assigned to each of the groups.
- Authentication modes allowed for users.
- Certificates needed for validating certificates presented by users who carry out authentication using a cryptographic card.
- Connections to the application databases and passwords for these connections.
- PKCS #11 library and parameters for accessing user cryptographic cards.
- PKCS #11 library and parameters for accessing cryptographic keys stored in a cryptographic device.
- References to the cryptographic keys used by the security functions (e.g. keys to sign KeyOne batches).
- References to the cryptographic keys used by the PKI services (e.g. certificates issuing keys).

5.5.2.1 Configuration Data Uses

The knowledge of configuration data allows KeyOne applications to:

- Authenticate users.
- Verify user access rights for different functions.
- Establish connections with the database.
- Access user cryptographic cards.
- Access cryptographic keys that the security functions (e.g. obfuscation keys, batch signing keys, i3D session signing keys) require.
- Access cryptographic keys that the PKI services require (e.g. certificate issuing keys).

5.5.3 Managing the KeyOne Configuration

This section breaks down the management of the KeyOne configuration by task.

5.5.3.1 Management of the Static Policy Parameters

There are only two management actions permitted for the static parameters:

- Selecting the system security policy (system initialization).

Replacing policy files (once the system is initialized).

5.5.3.2 Management of the System-Initialization Static Parameters

The value of the static parameters resulting from system initialization cannot be modified once defined (i.e., these parameters cannot be changed once the system is initialized).

5.5.3.3 Management of the Dynamic Configuration

Dynamic KeyOne configuration is managed by running administration sessions in KeyOne applications.

- The dynamic KeyOne system configuration is managed with the KeyOne Console application.
- The dynamic configuration of each KeyOne application instance is performed through the relevant administration application.

5.5.4 Protecting the KeyOne Configuration

This section explains how the different types of KeyOne configuration are protected.

5.5.4.1 Protecting the Dynamic Configuration

KeyOne dynamic configuration (both system configuration and application-specific configuration) is protected by a mechanism called obfuscation to ensure its confidentiality and integrity.

Obfuscation consists in encrypting data from the KeyOne configuration along with its digested value (hash) using an encryption key. This key (the obfuscation key) is obtained from secrets that are protected by being: hidden in the code (i.e., in the software), stored in a cryptographic device (hardware obfuscation mode), known to users with a specific role. CIMC security policy mode only supports the hardware obfuscation mode (the maximum level of protection for the KeyOne configuration).

5.5.4.2 Protecting the Static Configuration

The KeyOne static configuration is saved as clear text in policy files. The integrity and authentication of this information is assured by using digital signatures. Each policy file is signed using a key (whose verification certificate is contained inside the KeyOne binary code). When the system is started, the integrity and authenticity of this data is verified.

5.5.4.2.1 Protecting the Static Configuration Generated During Initialization

Some KeyOne static configuration data generated during initialization (e.g. database URI, security policy) are stored as plain text inside an installation file. An identical image of this file is saved in the dynamic configuration. This image is validated each time the system starts.

The rest of the static configuration generated during initialization is stored (protected by the obfuscation mechanism) in specific folders. Since this data is obfuscated, its integrity, authenticity and confidentiality are guaranteed.

5.5.5 Functional requirements satisfied by security functions

These services satisfy the following requirements:

FMT_SMF.1.1, FMT_MSA.1.1 (iteration 1), FMT_MSA.1.1 (iteration 2), FMT_MSA.3.1, FMT_MSA.3.2, FIA_ATD.1.1

El TOE is able to manage the security parameters and the security functions, using the protection mechanisms identified in the previous sections. Access control used in the management of security attributes and in the management of security functions is described in the Access Control Management, page 141.

5.6 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,...), or by means the use of KeyOne proprietary protocols (e.g. KeyOne batches, NDCCP protocol, ...).

5.6.1 KeyOne batches

Certificate generation and revocation services involve communication between a KeyOne Registration Authority 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 transactions between KeyOne Registration Authorities and KeyOne CA 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.

KeyOne Registration Authorities send certification or revocation requests to the KeyOne CA application by using the KeyOne batch format. The KeyOne CA application sends the generated certificates or the revocation results to the KeyOne Registration Authorities by using the KeyOne batch format.

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 make sure that the batch has not been modified during the transmission. Making use of the user data, the CA verifies that the entity sending the batch is the same as that signed that lot.
- 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.

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.

5.6.2 Functional requirements satisfied by security functions

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

- Batch Signature Function. 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. This functionality covers the validation by the KeyOne CA of a received KeyOne batch from the Registration Authority. 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.
- SSL/TLS between KeyOne Components. 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 Registration Authorities and KeyOne CA.
- Obfuscation Function. 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:

FDP_ITT.1.1 (iteration 1)

The FDP_ITT.1.1 (iteration 1) requirement needs the integrity service applied to the user data. The user data can be included in the communications between a Registration Authority and the KeyOne CA component (e.g. register information, ...).

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, KeyOne CA only accepts data contained in a signed batch. This signature must be generated by an authorised RA administrator.

The communications between the Registration Authority and the KeyOne CA uses the SSL/TLS secure protocol (with client authentication) in order to provide of the integrity service to these communications.

FDP_ITT.1.1 (iteration 2)

The FDP_ITT.1.1 (iteration 2) requirement needs the confidentiality service applied to the user data. The user data can be included in the communications between a Registration Authority and the KeyOne CA component (e.g. register information, ...).

The communications between KeyOne XRA and KeyOne CA uses the SSL/TLS secure protocol (with client authentication) in order to provide of the confidentiality service to these communications.

FDP_SDI_CIMC.3.1

The public key stored within the TOE, 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.
  - 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

FCO_NRO_CIMC.3.1, FCO_NRO_CIMC.3.2

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 KeyOne XRA and KeyOne CA involves information about the certificate status (revocation request from the KeyOne Registration Authority), 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 XRA. In this case the *casignature* field of the batch contains the batch detached signature generated by KeyOne CA, and KeyOne CA verifies the signature contained in the batch generated by the RA.

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 Registration Authority and the KeyOne CA components are stored in the batch table of the KeyOne database.

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.

Regarding communication between KeyOne VA and the user of OCSP service, the protection mechanisms provided are the OCSP message and the SSL security protocol. Regarding the OCSP response message, it is digitally signed, as is specified in [RFC2560]. About the SSL security protocol, KeyOne VA product can be configured so that it requires the use of SSL with client authentication in the OCSP communications

FCO_NRO_CIMC.3.3

FCO_NRO_CIMC.3.3 requires the verification of the evidence of origin of information for all security-relevant information.

Before processing a certification/revocation request, the KeyOne CA verifies the evidence generated by KeyOne XRA. 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.

FCO_NRO_CIMC.4.1

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.

FCO_NRO_CIMC.4.2

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.

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.

FPT_ITC.1.1

The FPT_ITC.1.1 requirement needs the protection against unauthorised disclosure, of the data transmitted from the KeyOne system to a remote trusted IT product. The communications that are affected with this requirement are the 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 Database client is carried out by means physical protection applied to the host. Regarding to the communication between the Database client and the Database server, it can be protected by means the SSL protocol established between these components.

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.

FDP_UCT.1.1

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 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 Database client is carried out by means physical protection applied to the host. Regarding to the communication between the Database client and the Database server, it can be protected by means the SSL protocol established between these components.

FPT_ITT.1.1 (iteration 1)

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 1) requirement needs the integrity service applied to the TSF data. The TSF data can be included in the communications between the KeyOne XRA and the KeyOne CA components.

In order to protect the TSF data in the communication between these components, KeyOne CA only accepts requests that are signed by an authorised administrator.

The communications between a Registration Authority and the KeyOne CA uses the SSL/TLS secure protocol (with client authentication) in order to provide of the integrity service to these communications.

FPT_ITT.1.1 (iteration 2)

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 2) requirement needs the confidentiality service applied to the TSF data. The TSF data can be included in the communications between the KeyOne XRA and KeyOne CA components.

The communications these components uses the SSL/TLS secure protocol (with client authentication) in order to provide of the confidentiality service to these communications.

FDP_CIMC_BKP.2.1

The FDP_CIMC_BKP.2.1 requirement needs that the backup data shall be protected against modification through the use of digital signatures, keyed hashes, or authentication codes.

The KeyOne System has backup functionality for backing up the whole KeyOne system required to reconstruct the current status from the backup and a copy of the same version of the software used to initial install the KeyOne system. The data stored in the system backup required to recreate the state of the system at the time the backup includes all the necessary information stored on the hard disk of the machine where the KeyOne system is installed. This information is protected by the TSF that protects this data from unauthorised modifications.

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 shall be stored in encrypted form only.

The KeyOne System has backup functionality for backing up the whole KeyOne system required to reconstruct the current status from the backup and a copy of the same version of the software used to initial install the KeyOne system. The data stored in the system backup required to recreate the state of the system at the time the backup includes all the necessary information stored on the hard disk of the machine where the KeyOne system is installed and all the information stored on the KeyOne Databases. This information is protected by means the TSF that protects these data from unauthorised disclosure using an encryption process.

5.7 Certification and keys Management

An important part of the KeyOne system is all the management of certification and keys 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.

5.7.1 Functional requirements satisfied by security functions

The Certification Management services are composed of the following security functions:

- Certification Request Verification Function. 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. 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.

- CRLs Generation Function. 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. 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[12]. This functionality is in charge of providing to the KeyOne CA of the functionality in order to manage revocation profiles by an authorised Administrator.

These services satisfy the following requirements:

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 TSF verifies the self-signed certification request that is included in a certification KeyOne batch.

FDP_CIMC_CER.1.1

The Certificates Generation Function 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.

FDP_CIMC_CER.1.2

Before the generation of the X.509 certificate, the TSF assures that the generated certificates are consistent with the currently defined certificate profile. Consequently TOE covers the accomplishment of the FDP_CIMC_CER.1.2 requirement.

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 TOE 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

[12] The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".

the signature field in is format, then the TSF covers the accomplishment of the FDP_SDI_CIMC.3.1 requirement.

FMT_MOF_CIMC.3.1

Before the generation of the X.509 certificate, TSF assures that the generated certificates are consistent with the currently defined certificate profile. Consequently the TOE covers the accomplishment of the FMT_MOF_CIMC.3.1 requirements.

FDP_CIMC_CER.1.4

The TSF 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 TOE checks these restrictions, then this function covers the accomplishment of the FDP_CIMC_CER.1.4 requirement.

FDP_CIMC_OCSP.1.1

The TSF checks the following restrictions in the generation of OCSP basic responses:

- The version field shall contain a 0.

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.

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

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

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

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

FDP_CIMC_CRL.1.1

The CRLs Generation Function 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.

FMT_MOF_CIMC.5.1

The CRLs Generation Function 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 TSF assures that the issued CRL is consistent with the certificate revocation list profile. Consequently the TOE covers the accomplishment of the FMT_MOF_CIMC.5.1 requirement.

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, the 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.

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 generation and it consists of signing the certificate.

The certificate 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 4211.

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).

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.

FMT_MOF_CIMC.3.2, FMT_MOF_CIMC.3.3, 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, KeyOne CA provides of a certification template mechanism.

FMT_MOF_CIMC.5.1

This requirement implies the functionality that allows managing a revocation profile[13] by the authorised Administrator. Through the Revocation Profile Function, the 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.

[13] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

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.

FMT_MOF_CIMC.6.1, FMT_MOF_CIMC.6.2, FMT_MOF_CIMC.6.3

In KeyOne VA application, OCSP responses can be configured by an Administrator. This specific rol can specify the set of acceptable values for the OCSP response fields. KeyOne VA generates OCSP responses considering the configuration of these responses.

FMT_MOF_CIMC.5.2

This requirement implies the functionality that allows managing a revocation profile[14] by the authorised Administrator. Through the Revocation Profile Function, the KeyOne CA provides of a revocation template mechanism.

FMT_MOF_CIMC.5.3

This requirement implies the functionality that allows managing a revocation profile[15] by the authorised Administrator. Through the Revocation Profile Function, the KeyOne CA provides of a revocation template mechanism.

FMT_MTD_CIMC.4.1

The TOE private keys are stored in a FIPS 140-2 level 3 validated cryptographic module.

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.

FMT_MTD_CIMC.7.1, FDP_ETC_CIMC.5.1

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.

FDP_ACF_CIMC.2.1, FDP_ACF_CIMC.2.2, FDP_ACF_CIMC.3.1

TOE personnel private keys are stored in FIPS 140-1 validated cryptographic module. The TOE does not store neither certificate subject private keys nor user secret keys.

FCS_CKM_CIMC.5.1

The TOE does not maintain plaintext secret and private keys.

[14] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

[15] *The concept of "revocation profile" used in this document, has been translated to the TOE (software and manuals) as "CRL template".*

5.8 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 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 (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).

5.8.1 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).

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.

5.8.2 Functional requirements satisfied by security functions

The Private Secure Store Access Function 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:

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 TSF 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 TSF.
- 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.

5.9 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 recovery functionality only can be invoked by the System Administrator role, and only this role can configure the parameters involved in this functionality.

5.9.1 Functional requirements satisfied by security functions

The TSF involves tasks related to the Backup and Recovery functionality located in the KeyOne system.

These services satisfy the following requirements:

FDP_CIMC_BKP.1.1, FDP_CIMC_BKP.1.2, FDP_CIMC_BKP.1.3

The FDP_CIMC_BKP.1.1 requirement requires that the TOE provides a backup function. The TSF implements a continuous backup process.

The objective of the backup process is to make sure the system has, at all times, all the information needed to invoke a recovery operation in case of failure. The backup operation takes into account all system data not covered by the backup functionality of the environment. Backup operation is continuous and takes part every time data is generated in the system that is not covered by an environment backup function.

FDP_CIMC_BKP.1.4

The FDP_CIMC_BKP.1.4 requirement requires that the TOE provides a recovery function capable of restoring the state of the system from a backup. The TSF implements functionality related to the recovery process. This restore process reconstructs 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 initially install the KeyOne system.

6 Bibliography, Definitions and Acronyms

6.1 Bibliography

The following documents are referenced in this document:

Reference	*Referenced document*
[CIMC]	Certificate Issuing and Management Components Family of Protection Profiles, Version 1.0. October 31, 2001. National Security Agency (NSA).
[ITU-T X.509v3]	X509v3: ITU-T Recommendation X.509 | ISO/IEC International Standard 9594-8, Information Technology – Open Systems Interconnection – The Directory: Authentication Framework
[CWA-14167-1]	CWA 14167-1: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements; CEN WORKSHOP AGREEMENT; June 2003.
[CC_31_Part1]	Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. September 2012, Version 3.1, Revision 4. CCMB-2012-09-001.
[CC_31_Part2]	Common Criteria for Information Technology Security Evaluation. Part 2: Security functional components. September 2012, Version 3.1, Revision 4. CCMB-2012-09-002.
[CC_31_Part3]	Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance requirements. September 2012, Version 3.1, Revision 4. CCMB-2012-09-003.
[TS101862]	ETSI Technical Specification. ETSI TS 101 862 V1.1.1 (2004-03). X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons.
[EUROPEAN_DIRECTIVE]	DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 13 December 1999 on a Community framework for electronic signatures.
[WEBTRUST]	www.webtrust.org

Reference	*Referenced document*
[ETSI_TS_101_456]	ETSI TS 101 456. Policy requirements for certification authorities issuing qualified certificates. ETSI.
[ETSI_TS_101_042]	ETSI TS 101 042. Policy requirements for certification authorities issuing public key certificates. ETSI.
[ETSI_TS_102_023]	ETSI TS 102 023. Electronic Signatures and Infrastructures (ESI); Policy requirements for time-stamping authorities. ETSI.
[ETSI_TS_101_862]	ETSI TS 101 862. Qualified certificate profile. ETSI.
[ETSI_TS_102_280]	ETSI TS 101 862. X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons. ETSI.
[ICAO_9303]	Document 9303 – Machine Readable Travel Documents. International Civil Aviation Organization.
[RFC3280]	RFC 3280. Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List (CRL) Profile. IETF.
[RFC5280]	RFC 5280. Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List (CRL) Profile. IETF.
[RFC3039]	RFC 3039. Internet X.509 Public Key Infrastructure. Qualified Certificates Profile. IETF.
[RFC3739]	RFC 3739. Internet X.509 Public Key Infrastructure. Qualified Certificates Profile. IETF.
[RFC5652]	RFC 5652. Cryptographic Message Syntax (CMS). IETF.
[RFC4210]	RFC 4210. Internet X.509 Public Key Infrastructure. Certificate Management Protocol (CMP). IETF.
[RFC4511]	RFC 4511. Lightweight Directory Access Protocol (LDAP): The Protocol. IETF.
[RFC4211]	RFC 4211. Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF). IETF.
[RFC2560]	RFC 2560. X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP.IETF.
[RFC6960]	RFC 2560. X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP.IETF.
[PKCS#1]	PKCS #1: RSA Cryptography Standard. RSA.
[K140CSO]	KeyOne 4.0 - Considerations for a Secure Operation. Document code: A5C273C9. Version 1.14.
[K140PIU]	Product Installation and Uninstallation. Document code: A98558AB. Version 1.45.
[K140UM]	User Manual. Document Code: 8B4B9CFE. Version 1.61.

Reference	*Referenced document*
[K140AU]	KeyOne Console 4.0 Administration and Use. Document code: 3999D586. Version 2.3.

6.2 Definitions

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 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: 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. Basically is an electronic attestation that links signature-verification data to a person and confirms the identity of that person.

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-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.

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;

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).

Security Function Policies (SFP): Rules that the TOE must enforce.

Security policy: The set of rules lay down by the security authority governing the use and provision of security services and facilities.

Subscriber: An entity subscribing with a CSP to have its public key and identity certified in a public key certificate.

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.

6.3 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
IT	Information Technologies
LRA	Lightweight Registration Authority
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NDCCP	Near Domain Cert-Status Coverage Protocol

Acronym	*Meaning*
OCSP	Online Certificate Status Protocol
OSP	Organisational Security Policies
PKI	Public key Infrastructure
PP	Protection Profile
RA	Registration Authority
SFP	Security Function Policies
SFR	Security Functional Requirements
SCD	Signature-Creation Device
SSCD	Security Signature Creation Device
TOE	Target Of Evaluation
TSF	TOE Security Function
TSP	Time-Stamp Protocol
VA	Validation Authority

SAFELAYER

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