IDEMIA CA v1.3.1

Public

Security Target

Reference: FQR 151 007-403

DOCUMENT EVOLUTION

Version/Edition	Issue Date	Author	Purpose
1	2025/09/04	IN Smart Identity	Document creation based on FQR 550 0219 Ed11.

Table of Contents

Table of figures

Table of tables

1 Generalities

1.1 Subject of the document

This document is the Public Security Target for the Common Criteria certification level EAL4 augmented with ALC_FLR.1 of the product « **IDEMIA CA v1.3.1**» developed by **IN Smart Identity**.

This document is based on the final version of the Security Target document FQR 550 0219 Ed 11.

1.2 Technical Terms

Term	Definition
Administrator	A person who performs TOE initialization, TOE personalization, or other TOE administrative functions.
Authentication data	Information used to verify the claimed identity of an operator and user.
Certificate	Entity information accompanied with a cryptographic public key, the whole being digitally signed to bind the entity information and the public key and identify that entity as legitimate holder of the public key.
Certification Authority	Trusted entity responsible for digitally signing and issuing certificates to other Certification Authorities of End-Entities.
Certificate Request	Also denoted as Certificate Signing Request (CSR), designates a certificate that, instead of being signed by a CA, is self-signed.
End-Entity	Person, organization or entity that is the subject of a leaf certificate or CSR in the TOE. Owner of a leaf certificate in the PKI, as opposed to a CA certificate.
Frontend	As opposed to "backend", designates the user interface software component. A frontend presents data to the user and relies on the TOE to process data and user requests.
User	Generally, a person operating the frontend in either CA Operator, RA Operator, or Auditor role. Likewise, an integrating application.

Term	Definition
Integrating Application	IN SMART IDENTITY -developed applications that interact with the TOE's REST API.
Intermediate CA	Intermediate Certification Authority, a certification authority that is signed by a Root CA or another Intermediate CA, and that issues certificates for other Intermediate CAs or End-Entities
Registration Authority	Assumes the role in charge of verifying the contents of a certificate request in a PKI, to validate the certificate information before they are signed by the CA.
Root CA	Root Certification Authority, anchor of trust in a PKI. Holds a self-signed certificate that is, by definition, assumed trusted.
Self-Signed (Certificate)	A certificate that is signed by its own private key.
Web Services	Alias for "Integrating Application"
X509 (or X.509)	Standard defining the format of public key certificates

1.3 Abbreviations

Acronym	Definition
ANSSI	*(French)* Agence Nationale de la Sécurité des Systèmes d'Information
API	*Application Programming Interface*
CA	*Certification Authority*
CC	*Common Criteria*
CIMC	*Certificate Issuing Management Component*
CIMS	*Certificate Issuing and Management System*
CP	*Certificate Policy*
CPS	*Certification Practices Statement*
CRL	*Certificate Revocation List*
CSR	*Certificate Signing Request*
CVC	*Card Verifiable Certificates*
CVCA	*Country Verifying Certification Authority* (equivalent of "Certification Authority" in EAC PKI)
DV	*Document Verifier*
EAC	*Extended Access Control*
EAL	*Evaluation Assurance Level*
FIPS	*Federal Information Processing Standards*
HSM	*Hardware Security Module*
IP	*Internet Protocol*
IS	*Inspection System* (equivalent of "End Entities" in EAC PKI)

Acronym	Definition
JSON	*JavaScript Object Notation*
JWT	*JSON Web Token*
OCSP	*Online Certificate Status Protocol*
OE	*Objective for the Environment*
OSP	*Organizational Security Policy*
OT	*Objective for the TOE*
PKCS	*Public Key Cryptography Standards*
PKI	*Public Key Infrastructure*
PP	*Protection Profile*
RA	*Registration Authority*
RGS	*(French)* Référentiel Général de Sécurité
SAR	*Security Assurance Requirement*
SFR	*Security Functional Requirement*
ST	*Security Target*
TOE	*Target Of Evaluation*
TSF	*TOE security functionality*

1.4 References

Ref.	Document Title
[BSI-TR03110]	"Technical Guideline TR-03110 Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token - Part 3: Common Specifications". Federal Office for Information Security (German, BSI), version 2.21, 21 December 2016.
[CCPart1]	"Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model". CCMB-2017-04-001. April 2017, Version 3.1 Rev.5
[CCPart2]	"Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements". CCMB-2017-04-002. April 2017, Version 3.1 Rev.5
[CCPart3]	"Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance requirements". CCMB-2017-04-003. April 2017, Version 3.1 Rev.5
[CrySpec]	"FQR 220 1783 Ed 7 - IDEMIA CA - Cryptographic Specifications", Ed 7, AMOSSYS and IN SMART IDENTITY, 17/04/2025
[ICAO9303]	"Doc 9303 – Machine Readable Travel Documents". International Civil Aviation Organization, 2015, 7th edition.
[OIDC]	"Open ID Connect Specifications". OpenID Connect Working Group, November 2014. Version 1.0 errata 1. https://openid.net/specs/openid-connect-core-1_0.html
[PP]	"Certificate Issuing and Management Components Protection Profile", NIST, version 1.5, 11/08/2011
[RFC5280]	"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, IETF NetworkWorking Group, May 2008.
[NT_CRYPTO]	"Guide des mécanismes cryptographiques - Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques", ANSSI, version 2.04 du 01/01/2021
[RGS_B2]	"Gestion des clés cryptographiques : Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques", ANSSI, Annexe B2 version 2.00 du 13 juin 2012

Ref.	Document Title
[RGS_B3]	"Authentification, Règles et recommandations des mécanismes cryptographiques de niveau de robustesse standard", réf. 729/SGDN/DCSSI/SDS/AsTeC, version 1.10
[AGD_PRE]	FQR 220 1622 Ed 7 - IDEMIA CA - AGD_PRE
[AGD_OPE]	FQR 220 1654 Ed 6 - IDEMIA CA - AGD_OPE
[SOLS_MAN]	FQR 220 1616 Ed 9 - IDEMIA CA - Solution Manual
[ADMIN_GUIDE]	FQR 220 1615 Ed 10 - IDEMIA CA - Admin Guide
[ICD]	FQR 220 1618 Ed 8 – IDEMIA CA - ICD

2 ST Introduction

2.1 ST Reference and TOE reference

2.1.1 ST Reference

Title	IDEMIA CA v1.3.1 Public Security Target
ST Reference	FQR 151 007-403
ST Version	Ed 1
CC Version	3.1 Revision 5
Assurance Level	EAL4 augmented with ALC_FLR.1
ITSEF	AMOSSYS
Certification Body	ANSSI
Author	IN Smart Identity
Publication Date	04/09/2025

2.1.2 TOE Reference

TOE Commercial Name	IDEMIA CA
TOE Version	1.3.1
Developer Name	IN Smart Identity
Link to the developer	https://www.ingroupe.com/
Applet Code Version (SAAAAR Code)	SAAAAR Code: 203673
Guidance Documents	[AGD_PRE], [AGD_OPE]

2.1.3 TOE Identification

The TOE provides a REST API Endpoint that provides the TOE identification (URL `/pki-ca/actuator/info`). This endpoint is accessible for all authorized user of the TOE.

The details provided by the endpoint are:

- the name of the TOE - IDEMIA CA,
- the version of the TOE – 1.3.1,
- the code for the TOE – 203673,
- the build number of the TOE which consists of the TOE version and the first 6 character of the SHA-1 hash of the last commit,
- the Java Version on which the TOE is running on,
- The certificate type of which the TOE is configured – X509 or CVC,
- The mode for which the TOE is configured – see mode type listed in Section 2.3.2.

2.2 TOE Overview

The TOE overview summarizes the usage and major security features of the TOE.

Note that this ST is inspired from (but *does not* claim conformance to) the "*Certificate Issuing and Management Components Protection Profile*" [PP].

2.2.1 TOE Type

The TOE, IDEMIA CA, is the core component of the Citizen PKI application developed by IN SMART IDENTITY. It is thus a subset of this application.

The TOE is purely software: it is a server application developed in Java. It manages public key certificates of Certification Authorities (CA), issues certificates to its clients, and puts in place a Public Key Infrastructure (PKI). The TOE covers both PKI for X509 Public Key Certificates, and Card Verifiable Certificates (CVC) for Extended Access Control (EAC).

2.2.2 Required non-TOE hardware/software/firmware

See section 2.4 for the exact list of hardware and software dependencies.

The TOE is configured in a Docker image and runs as a Docker container.

The TOE is a *Certification Authority Web Service Application* and thus does not have any interactive user interface. A web application, also developed by IN SMART IDENTITY, serves as a frontend component that is responsible for interfacing with the web users, and forwarding web requests as REST API requests to the backend web service application. There can also be *integrating applications* (developed by IN SMART IDENTITY or by third parties) that use the REST API of the web service application. The web application is not included in the TOE.

2.2.3 Usage and major security features

The Citizen PKI application is used to manage Certification Authorities (CA) or Public Key Infrastructures (PKI). The TOE can be configured to manage either X509 [RFC5280] or CVC [BSI-TR03110] in a single setup.

The produced end entity certificates, and more generally, the certification chains, are meant to be used inside an organization or system infrastructure to establish trust between actors or devices, for instance to enable secure communications, or to authenticate individuals or devices.

The TOE is a subset of the IN SMART IDENTITY Citizen PKI CA application.

2.3 TOE Description

This section provides a description of the TOE and its environment. The description starts with a generic introduction to PKIs.

2.3.1 Introduction to Public Key Infrastructures

Note: this section is inspired from the presentation of PKI in [PP]. It is a generic PKI management application description.

A PKI is a security infrastructure that creates and manages public key certificates to facilitate the use of public key cryptography. To achieve this goal, a PKI must perform two basic tasks:

- generate and distribute public key certificates to bind public keys to other information (subject name, public key intended usage, expiration date, etc.) after validating the accuracy of the binding,
- maintain and distribute certificate status information for unexpired certificates.

Certificate types. The TOE manipulates two kinds of certificates: X509 certificates and *Card Verifiable Certificates* (CVC). The formers are the most common type of certificates.

PKI Structure. Depending on the type of certificates involved, the structure of PKIs can vary. Figure 1 shows a possible hierarchy for X509 certificates, and Figure 2 illustrates the case for CVC. The latter usually involve a simpler structure than X509 certificates, so that the verification of a certificate chain is made simpler. The *Country Verifying Certificate Authority* (CVCA) is the equivalent of the X509 root CA; the *Document Verifier Certificate Authority* (DVCA) is the equivalent of an intermediate CA; and the *Inspection System* (IS) is the equivalent of the leaf/end entity.

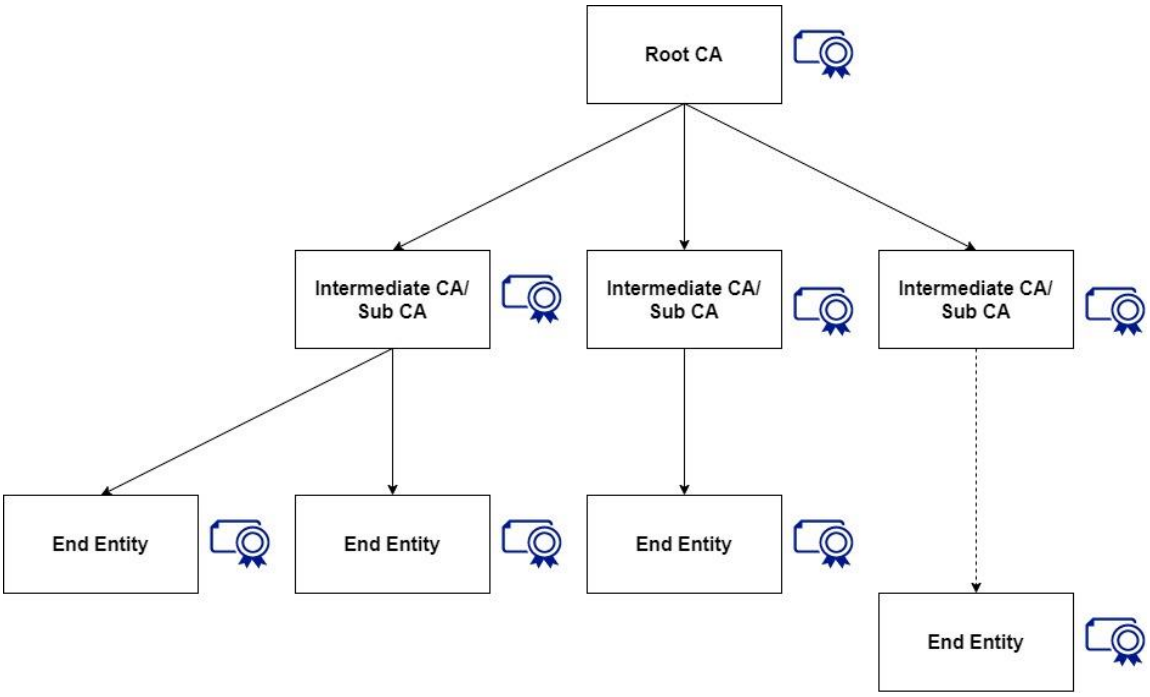

Figure 1 - Example of X509 PKI Hierarchy

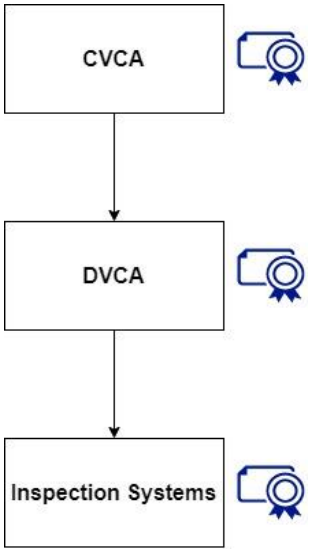

Figure 2 - Example of Extended Access Control PKI Hierarchy

Certificate revocation status. In a PKI, the *revocation status* of certificates must be maintained. For X509 PKIs, certificates have the status *valid* when they are emitted. When the certificate (or its associated key pair) is stolen or re-emitted, *revocation status* evolves: it may become *compromised* or *superseded*. A PKI must be able to maintain and distribute accurate information on the status of the certificates it emitted.

PKI Roles. Operating a PKI requires at least the following components, or roles:

- *Certification Authorities (CA)*: operators in charge of managing the CAs, signing the certificates, and maintaining the certificates statuses,
- *Registration Authorities (RA)*: operators in charge of verifying the information in the public key certificates, before their signature by CA Operators.

Other roles may include *auditors*, operators in charge of verifying the log of actions performed by CA and RA Operators.

In accordance with the *Certificate Issuing and Management Components Protection Profile* [PP] the aggregation of CAs, additional components performing core tasks, and the personnel and procedures of operation will be denoted as *Certificate Issuing and Management System (CIMS)* in this document.

2.3.2 Generalities about the Citizen PKI application and about the TOE

The Citizen PKI is a product developed by IN SMART IDENTITY allowing management of PKIs. The TOE is a subset of this application. To place the TOE in its context, this section presents the whole Citizen PKI application and then specifies the boundaries of the TOE inside this application.

2.3.2.1 PKI Functionalities

With respect to the generic description of PKI management applications in the previous sections, the Citizen PKI application allows the management of certification authorities, the issuance and management of certificates, the production and maintenance of CRLs, and it is suited to manage hierarchical PKIs as well as interface with other existing PKIs. It manages both X509 and EAC certificates.

The TOE is designed to act as any node of the PKI hierarchy. Figure 3 and Figure 4 show where the Citizen PKI CA application can intervene in an X509 and EAC PKI respectively.

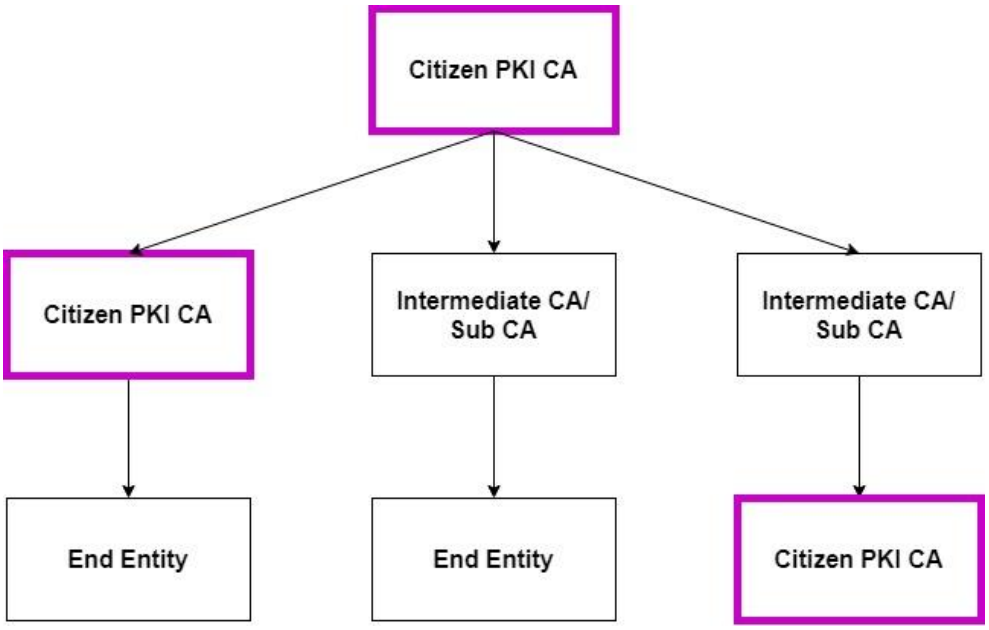

Figure 3 - Citizen PKI application as X509 PKI Nodes.

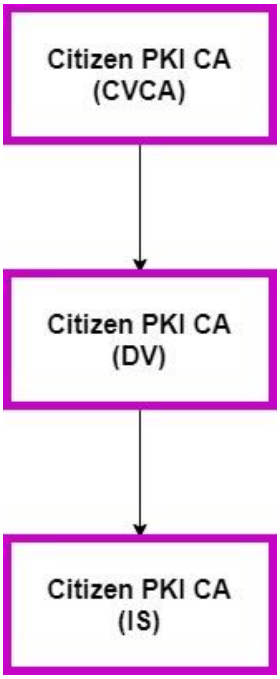

Figure 4 - Citizen PKI application as EAC PKI Nodes.

2.3.2.2 Modes of the TOE (PKI Roles)

The TOE can be configured to act as a specific node of a PKI:

- **Mode "Root CA"** – This mode is the default mode of the TOE. This mode is configured for the TOE to be only able to generate Root CA certificates and link CA certificates. It can also accept CSRs and issue certificates to its clients.

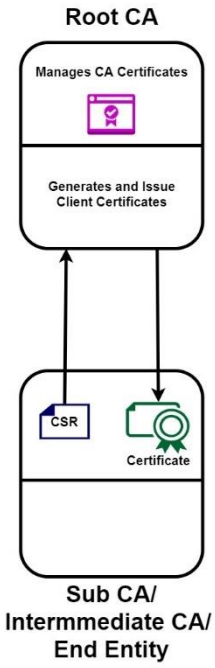

Figure 5 - Certificate handling and issuance for Root CA Node

- **Mode "Intermediate CA"** – This mode is configured so that the TOE would only be able to generate CSRs for its CA certificates and must rely on a root CA to issue its certificate. Being a sub CA, the TOE is able to accept CSRs and issue client certificates.

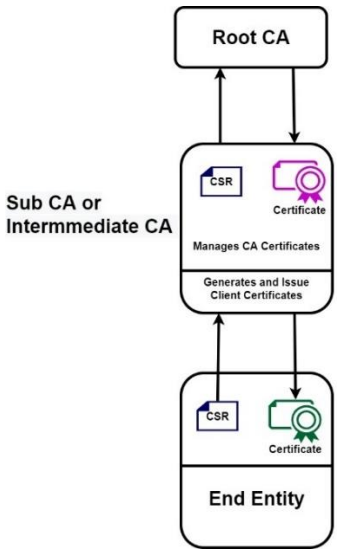

Figure 6 - Certificate handling and issuance for Sub CA Node

- **Mode "End Entity"**- This mode is configured so the TOE would only be able to generate CSRs for end entities and receive signed certificates from a root or sub CA. Being an end entity, the TOE is no longer able to accept and issue client certificates. The TOE may however use the *generate signature* module to make use of its private keys.

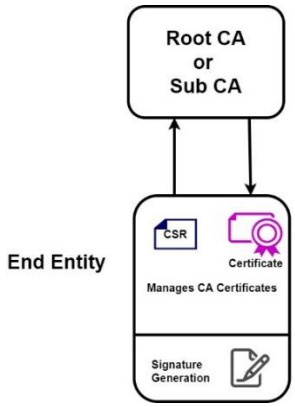

Figure 7 - Certificate handling for End Entity Node

2.3.2.3 Software Architecture

The TOE is composed of IDEMIA CA Parent and IDEMIA Crypto Core. These components contain the different application layers and cryptographic implementations of the TOE respectively. The subsystem and interactions are depicted in Figure 8, Software Architecture.

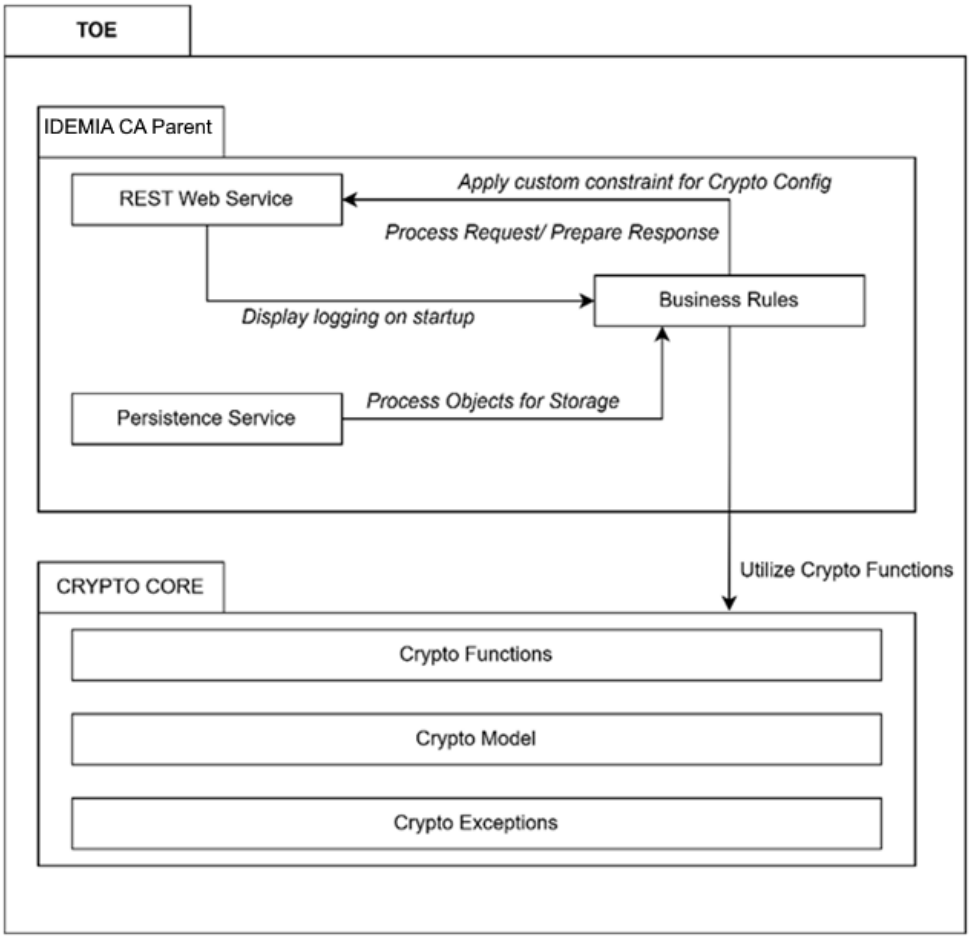

Figure 8 - Software Architecture

2.3.2.4 Possible Configurations

The Citizen PKI application can be installed in different configurations, depending on the *mode* and on the *certificate type* considered.

The different modes pertain to the PKI role set for the TOE in a PKI: root CA, Intermediate CA, and End Entity.

The different certificate types are X509 certificates and CVC.

Teams putting up the PKI can have several TOE containers depending on the PKI structure designed and identified.

2.3.3 Users

Users of the TOE can have one among four user roles, i.e., CA Operator, RA Operator, Auditor, and Integrating Application.

The user roles are communicated to the TOE in each REST API request. Section 4.1 Users and Roles provides a detailed discussion of user roles; for the processing of user roles, refer to Section 2.3.6.1 Access Control.

2.3.4 REST API

The TOE is made of the IDEMIA CA component. The application is accessible through an HTTP REST API. The frontend is responsible for transforming human operator's action on the web interface into REST API requests and presenting the response back to the user.

Access to the REST API is protected. Firstly, the HTTP connection is protected by TLS (see section 2.3.5). Secondly, the TOE refuses requests without a corresponding authentication token and responds with a *401 Unauthorized* response code. The frontend sends the user's session authentication token as the HTTP Bearer Token in each requests.

2.3.5 Communication channels

The TOE communicates with its dependencies and environment using various channels:

- The TOE requires TLS v1.3 for communication with integrating applications, database, and authentication servers
- the TOE and HSM communicate through proprietary Secure Messaging scheme of the HSM

2.3.6 Authentication and access control

The Citizen PKI application requires users to authenticate before using the functionalities and enforces access control between the four roles.

2.3.6.1 Access Control

The four user roles (CA Operator, RA Operator, Auditor, and Integrating Application) have different rights to view or modify data.

All API endpoints of the TOE have pre-defined set of user roles that grants corresponding access.

2.3.6.2 Session lifetime

A session in the TOE is determined by an access token.

2.3.7 Cryptographic operations and HSM

The TOE directly or indirectly performs various cryptographic operations: indirectly with the help of a HSM, or directly through various libraries.

The TOE depends on a hardware HSM to store CA private keys and perform cryptographic operations that involve CA key pairs. The HSM supported by the Citizen PKI application is the Utimaco IS GmbH v5.1 in FIPS 140-2 mode[5].

In addition to the HSM, the TOE also performs various cryptographic operations.

2.4 Physical scope

The TOE is software only and does not have hardware components.

As shown in Figure 10, the TOE is a software that runs in a Docker container. It can run on any 64-bit platform that is able to run Docker.

[5] The compliance with FIPS 140-2 cryptography is mentioned here for completeness. This security target aims for ANSSI-compliant cryptography [NT_CRYPTO].

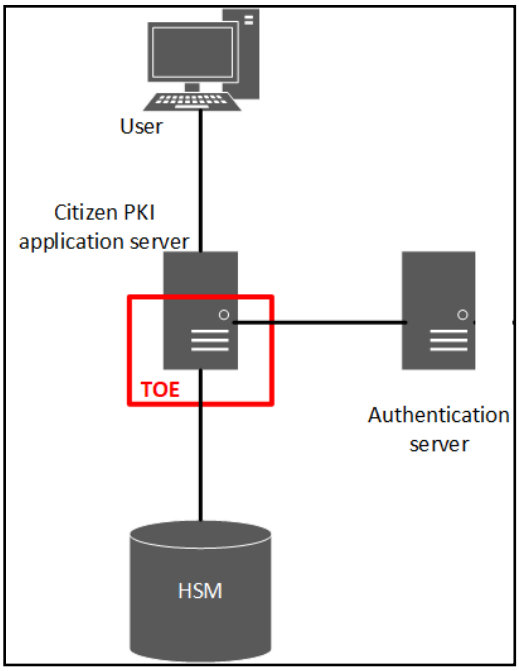

Figure 10 - Physical scope

2.4.1 TOE Component

The TOE is a part of the Citizen PKI application. The latter is made of the following components with the TOE in boldface:

- a Docker image for IDEMIA CA component server, containing:
  - base Docker image (Oracle Linux)
  - OpenJDK
  - **a Java JAR for the IDEMIA CA application, i.e., the TOE**
  - configuration files
  - HSM shared library and configuration
- a Docker image for the frontend component server
- Authentication Server
- HSM
- Database (High-Availability default configuration)
- documentation:
  - an administration [ADMIN_GUIDE],
  - a user guide [SOLS_MAN] and [ICD].

2.4.2 Delivery

The official build identifier (203673) of the TOE will be published on the IN SMART IDENTITY Nexus Product Repository. IN SMART IDENTITY Program Team will have to request access for the official build. Once access is granted, the Program Team will be allowed to download the

TOE. The Program Team will be provided access to the TOE *Solution Manual* [SOLS_MAN] and *Administration Guide* [ADMIN_GUIDE] for them to identify the setup they need for their program.

2.5 Logical scope

The logical scope of the TOE is depicted in Figure 11.

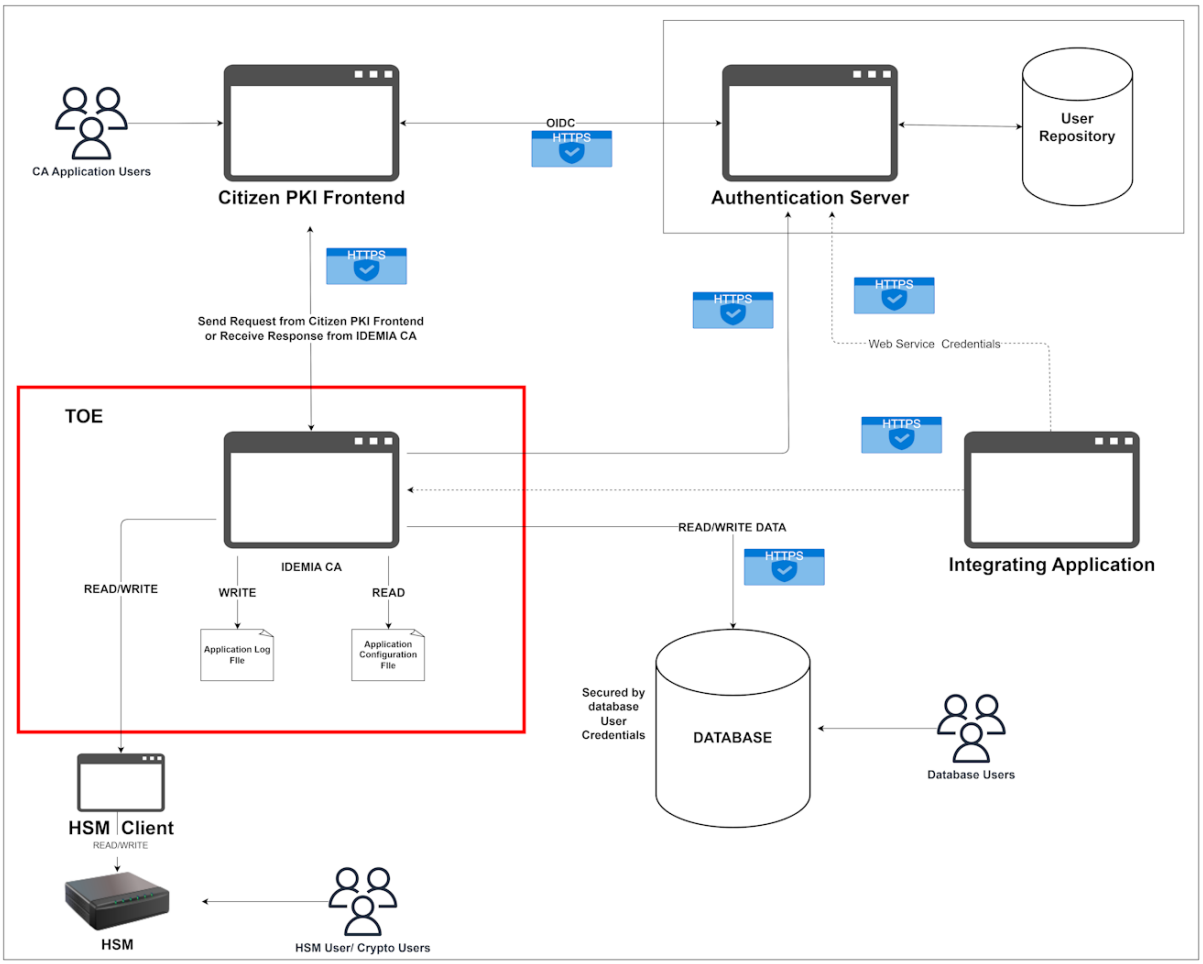

Figure 11 - Logical scope

2.5.1 Inclusions and exclusions of the logical scope

The TOE is the IDEMIA CA component.

The logical scope therefore implicitly includes:

- all logical features (including data import/export features) of the IDEMIA CA and HSM client modules,

- all certificate types, i.e. X509 and CVC,
- the communication channel from the IDEMIA CA component to the database
- the communication channel from the IDEMIA CA component to the HSM client module, and from the HSM client module to the HSM
- the communication channel from the frontend component to the IDEMIA CA component,
- the communication channel from the IDEMIA CA component to the authentication server,
- disk accesses to the audit log and configuration files,
- the communication channel from the integrating applications to the IDEMIA CA component

All other element is not part of the TOE. In particular, the frontend, which encompasses the web user interface, the authentication server, HSM client, and HSM hardware are *not* part of the TOE.

2.5.2 Logical scope defined in the terms of the PP

The logical boundaries of the TOE can also be defined using the terminology and concepts of the *Certificate Issuing and Management Components Protection Profile* [PP].

The TOE is a part of a *Certificate Issuing Management Component (CIMC)*, made of one or more *Registration Authorities* and several *Certification Authorities.* The boundaries of this CIMC is depicted in Figure 12; and the boundaries of the TOE inside this CIMC are depicted in Figure 13. These illustrations are adapted from the PP.

Figure 12 shows that the considered CIMC is mainly responsible for managing the CA(s) and RA(s). It manages CAs, receives certification requests (CSRs), and exports certificates and CRLs. The envisioned CIMC can optionally offer CRL or OCSP server, or other side-services such as a key recovery service for subject's keys, under the form of integrating applications.

Figure 13 shows that, inside this CIMC, only a subset of the functions are performed by the TOE. It manages CA generation and lifecycle, import requests (CSRs), signs them and exports certificates. It is responsible for producing the audit logs, but it is the OS that is responsible for storing them (and for enforcing access control over them). Similarly, the TOE verifies the authentication tokens, and considers the role included in a user's token, but it is the authentication server that is responsible for authenticating the user and creating the tokens. In terms of cryptography, the TOE performs minor cryptographic operations, but the bulk of cryptography is performed by the HSM. Every operation related to CA's private keys are performed in the HSM.

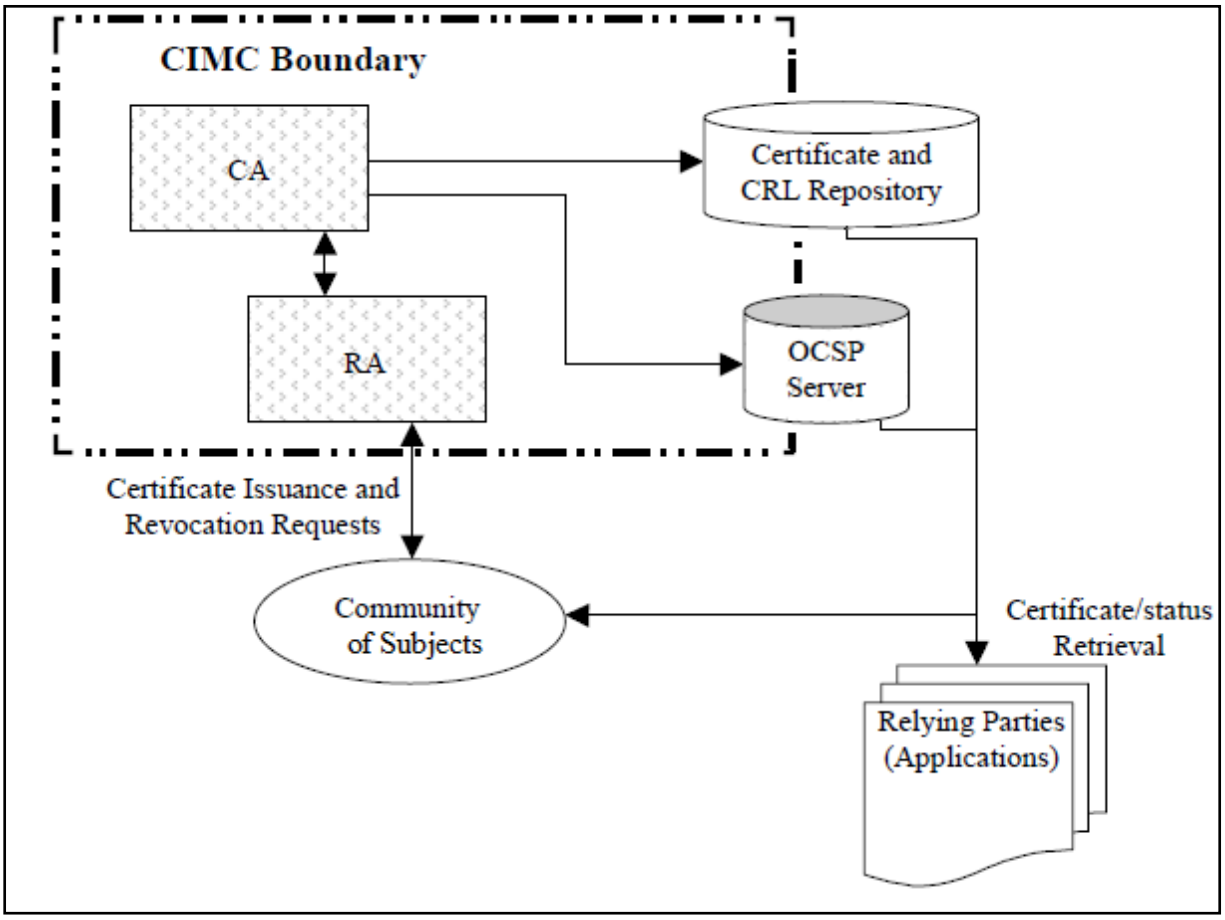

Figure 12 - Logical boundaries of the CIMC in the CIMS (adapted from [PP] Fig. 2)

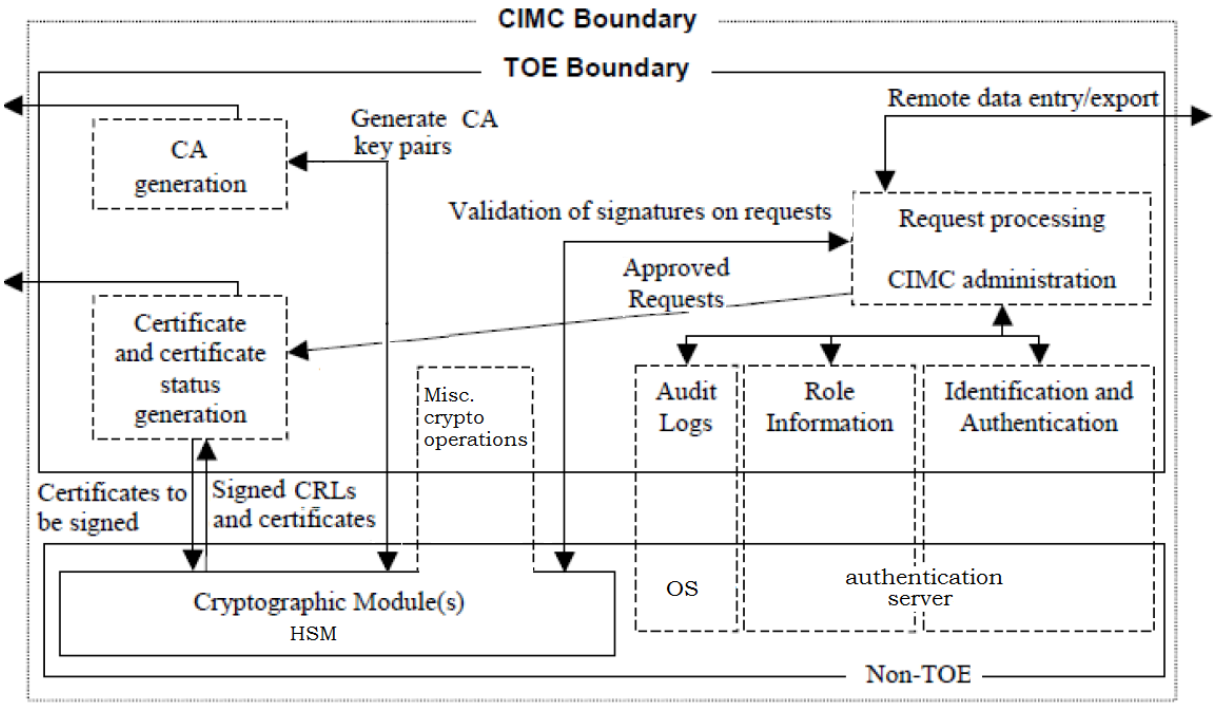

Figure 13 - Logical boundaries of the TOE inside the CIMC (adapted from [PP] Fig. 3)

Note that these CIMC and TOE boundaries respect the constraints imposed by the PP: the CIMC is only required to provide the CA and RA services, and the TOE is allowed to delegate cryptographic processing to a separate cryptographic module.

2.6 TOE environment

The TOE along with its dependencies is physically deployed in a secure facility that restricts access to authorized personnel.

The servers that are part of the Citizen PKI application are not publicly accessible in the Internet. They are deployed in a local network that is considered trusted.

3 Conformance Claims

3.1 Common Criteria Conformance

This security target (ST) claims conformance to the Common Criteria version 3.1, revision 5 [CCPart1] [CCPart2] [CCPart3].

This security target claims the following conformance:

- CC Part 2 Extended: the extended components of CC Part 2 are only from the protection profile [PP] presented below.
- CC Part 3 Conformant.

3.2 Security Requirement Packages

This ST claims conformance to the EAL4 assurance package, augmented with ALC_FLR.1, all defined in [CCPart3].

3.3 Protection Profile Conformance

This security target does not claim conformance to any PP.

It is inspired from "*Certificate Issuing and Management Components Protection Profile*" version 1.5 [PP]. This ST defines an SPD like that of the PP, and it reuses some arguments of the PP for the rationale of the tracing from Threats/Assumptions/OSPs to Security Objectives and SFR to Objectives for the TOE.

4 Security Problem Definition

4.1 Users and Roles

The frontend component of the Citizen PKI and integrating applications are the user of the TOE. Human operators and administrators accesses the TOE indirectly through the frontend component or integrating applications. While the latter directly access the TOE's interfaces (the REST API), the human users do not interact with the TOE directly. The frontend act as a simple forwarder of user's requests that transforms a UI action into the appropriate REST API endpoint call. The frontend acts on behalf of human operators/administrators.

There can only be one among four possible user roles in the TOE:

- **CA Operator**: manages CAs by creating, deleting, revoking them. More generally, CA operators perform all CA operations that necessitate the CA's private key: creation of CRLs, (cross-)certification of other CAs, and signature of client certificates. CA operators are the only users able to unlock the HSM
- **RA Operator**: is responsible for uploading certificate requests (CSRs) and validating them. A RA operator verifies the information embedded in the request and accepts or rejects it according to the Certification Policy. A RA Operator can also perform minor CA operations
- **Auditor**: the least privileged user role, only allowed to consult log entries and actions of other users,
- **Integrating Application**: assigned to integrating applications that interact with the TOE

Mapping of PP roles to the ST roles. Note that the [PP] defines four roles. The responsibilities of each role are described in section 5.2 of the PP, which also states that these roles and responsibilities can be distributed arbitrarily in conforming STs. Although this ST does not actually claim conformance, here are the correspondence between the roles in the PP and the above roles:

- System Administrators in the ST encompass the PP's Operators and Administrators (except for the configuration of profiles);
- CA Operator, RA Operator and Integrating Application in the ST together form the role of Officer in the PP (and the responsibility of the configuration of profiles is transferred to the Officers, i.e. CA Operators);
- and the Auditor role in the ST is the same as the Auditor role in the PP.

4.2 Assets

An asset is a piece of data (or a function) assessed to be of value for the TOE. Its value is estimated according to safety criteria (also called security needs): availability, integrity, confidentiality, and/or authenticity.

The sensitive assets protected by the TOE and its environment are **user data (DU)** or **TOE data (DT)** defined as follows.

User data	Properties	Definition
DU.HSM_PIN_CODE	Confidentiality	HSM PIN

TOE Data	Properties	Definition
DT.CA_PRIV_KEY	Confidentiality, Integrity	Private keys of CAs, used to sign certificates
DT.CA_CERT	Integrity, Authenticity	Certificates of CAs (includes the public key)
DT.CA_CRL	Integrity, Authenticity	CRLs produced by CAs
DT.CA_INTERNAL	Integrity	Internal state of CAs (serial numbers, certificates statuses, …)
DT.END_ENTITY_CERT	Integrity, Authenticity	Client certificates (including the end entity's public key), signed by a CA
DT.END_ENTITY_CSR	Integrity, Authenticity	CSR submitted by end entities (including the client's public key, its self-signature, and additional data)
DT.AUTH_TOKEN	Integrity, Authenticity	Authentication token delivered by the authentication server
DT.AUTH_SERV_VERIF_KEY	Integrity, Authenticity	Authentication server's public key
DT.SECURE_COMM_KEYS	Confidentiality, Integrity	Secret/Private keying material used by the TOE to establish secure channels
DT.SECURE_COMM_PUBLIC	Integrity	Public cryptographic material used by the TOE to establish secure channels.
DT.AUDIT_LOGS	Confidentiality, Integrity	Contents of audit log
DT.CONFIGURATION	Integrity	Contents of configuration files

4.3 Assumptions

An assumption is a statement on the context of use of the TOE or on the TOE environment.

4.3.1 Assumptions from the PP

The following assumptions are drawn from [PP].

- **A.AUDITORS_REVIEW_LOGS**

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

  From PP: *A.Auditor Review Audit Logs*

- **A.COMPETENT_OPERATORS**

  Competent Administrators, Operators, Officers and Auditors will be assigned to manage the TOE and the security of the information it contains. They require access to at least some of the information managed by the TOE and are expected to act in a cooperative manner

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

  From PP: merge of *A.Competent Administrators, Operators, Officers and Auditors*, *A.Cooperative User* (since users and operators are the same entity in this ST), *A.CPS*, and *A.Autentication Data Management*.

- **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).

  From PP: *A.Disposal of Authentication Data*

- **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.

  From PP: *A.Notify Authorities of Security Issues*

- **A.SOCIAL_ENGINEERING_TRAINING**

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

  From PP: *A.Social Engineering Training*

- **A.OPERATING_SYSTEM**

  The operating system has been selected to provide the functions required by this CIMC to counter the perceived threats identified in this ST.

  In particular, the hardware platform (PC or server) and OS on which the TOE is installed is supposed to be up-to-date, and free of malicious code at the moment of install.

  From PP: *A.Operating System*.

- **A.PHYSICAL_COMMUNICATIONS_PROTECTIONS**

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

  Note: this assumption only covers the availability *of the* physical *communication medium; it does not cover integrity, authenticity or confidentiality of the data in transfer.*

  From PP: physical protection part of *A.Communications Protections*

- **A.PHYSICAL_PROTECTIONS**

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

  From PP: *A.Physical Protection*

4.3.2 Additional Assumptions

This ST additionally introduces the following assumptions:

- **A.SANE_INSTALL**

  Those responsible for the TOE must adhere to IT security from delivery, installation, management, and operation.

- **A.HSM_SECURITY**

  The HSM used by the TOE is presumed secure and correct. It securely stores the cryptographic objects whereby unauthorized access or modification are not possible. It correctly performs cryptographic operations. Lastly, it has a strong random number generator used for key generation.

- **A.TRUSTED_AUTH_SERVER**

  The authentication servers on which the TOE relies to authenticate and identify users are trusted, managed by a trusted administrator. They are supposed to be secure and inaccessible from unauthorized entities.

- **A.DATABASE_SECURITY_AND_BACKUP**

  The data stored in the database (out of the TOE) is protected in integrity and confidentiality. Database administrators put in place the best practices in terms of monitoring, backup, and restoring of the database. The hardware on which the database runs is also assumed protected.

4.4 Threats

A threat consists of an adverse action performed by a threat agent an asset. Assets are defined in section 4.2, and the threat agents are:

- authorized users of the TOE defined in section 4.1 (in line with the [PP], they are considered as threats). Following the PP, they are often designated as "Administrators, Operators, Officers or Auditors".
- entities without legitimate access to the TOE, e.g., entities present in the local or remote network, or on the computing platform of the TOE. Sometimes denoted as *hacker* in the description of threats extracted from the PP.
- the *subjects* of the certificates, who do not directly interact with the TOE but may try to insert malicious payloads into certification requests.
- the *TOE developer,* the *CIMC hardware*, and the IT system (or IT personnel) also appear as threat agents in some threats of the PP.

4.4.1 Threats from the PP

The following first list of threats are drawn from [PP]. The description of threats define the entity, adverse action and asset. The text of their description was slightly adapted in some cases (for instance, since the TOE does not produce OCSP responses, the parts of threats mentioning OCSP responses were redacted). If text is added to the description, it is written in italic font.

- **T.CRITICAL_COMPONENT_FAILURE**

  Failure of one or more system components results in the loss of system critical functionality. Threat agent in this case is the CIMC hardware. Adverse action can be compromise of the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

  From PP: *T.Critical system component fails*

- **T.FLAWED_CODE**

  A system or applications developer delivers code that does not perform according to specifications or contains security flaws. Threat agent in this case is the TOE developer. Adverse action can compromise of the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

  From PP: *T.Flawed code*

- **T.MALICIOUS_CODE_EXPLOITATION**

  An authorized *user*, or hacker downloads and executes malicious code, which causes abnormal processes that violate the integrity, availability, or confidentiality of the system assets. Threat agent could be an authorized user or an unauthorized user.

Adverse action can be compromise of the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

From PP: *T.Malicious code exploitation*. Text modified to remove "the TOE itself" as a threat agent, since it does not make sense.

- **T.MESSAGE_CONTENT_TAMPERING**

A hacker modifies *or read* information that is intercepted from a communications link between two unsuspecting entities before passing it on to the intended recipient. Threat agent is an unauthorized user. Adverse action can be compromise of the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs*; or the disclosure of confidential data such as private/secret keys or credentials.*

From PP: this threat is *T.Message content modification* which was augmented to also include the threat of *reading* confidential message contents.

- **T.DISCLOSURE_OF_SECRET_KEYS**

A private or secret key is improperly disclosed. Threat agent is an authorized user*, an unauthorized entity,* or erroneous protocol. Adverse action can compromise the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

From PP: *T.Disclosure of private and secret keys*

- **T.MODIFICATION_OF_SECRET_KEYS**

A secret/private key is modified. Threat agent is an authorized user, *an unauthorized entity,* or erroneous protocol. Adverse action can compromise the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

From PP: *T.Modification of private/secret keys*

- **T.SENDER_DENIABILITY**

The sender of a message denies sending the message to avoid accountability for sending the message and for subsequent action or inaction. The threat agent is a subscriber to the CIMC *(i.e. a CA or certificate subject)*. Adverse action can reduce trust in CIMC.

From PP: *T.Sender denies sending information*

- **T.UNAUTHORIZED_ENTITY_ACCESS**

A hacker masquerades as an authorized user to perform operations that will be attributed to the authorized user or a system process or gain undetected access to a

system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality, or availability. Threat agent is the unauthorized user. Alternatively, an external entity gains access to the TOE by learning or stealing the credentials of an authorized user, to interact with the TOE. These credentials can be the HSM pin code, or authentication tokens (user login/password are only *implicitly* included, in the sense that with them, an attacker can obtain a valid authentication token, but login/password are never manipulated by the TOE). Adverse action can compromise the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

From PP: *T.Hacker gains access* augmented with credential theft

- **T.PHYSICAL_UNAUTHORIZED_ENTITY_ACCESS**

  A hacker physically interacts with the system to exploit vulnerabilities in the physical environment, resulting in arbitrary security compromises. Threat agent is an unauthorized user. Adverse action can compromise the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

  From PP: *T.Hacker physical access*

4.4.2 Additional Threats

This ST additionally introduces the following threats.

- **T.WEAK_CRYPTOGRAPHY**

  A weak hash, signature scheme or other cryptographic mechanism can be compromised by an attacker and used to apply integrity checks to malicious content so that it appears legitimate. The threat agent is an authorized user or an unauthorized entity. Adverse action can compromise the PKI objects such as certificates or CRLs.

- **T.RESIDUAL_MEMORY_ACCESS**

  An entity gets holds of sensitive values such as keys, passwords or credentials, by inspecting the TOE's memory during or after the manipulation of these sensitive values. Threat agents are the authorized users or unauthorized entities. Adverse action can compromise private keys or passwords, which in turn can be used to compromise of the security of the CIMC and/or relying party systems that rely on the PKI objects such as certificates or CRLs.

- **T.UNDETECTED_ACTIONS**

  Authorized users or unauthorized entities may take actions that adversely affect the security of the TOE, but that are not detected by the TOE or Auditors. The adverse action is the fact of evading detection, and the assets targeted are the audit logs.

- **T.CERTIFICATE_CORRUPTION**

  The certificate of a CA (root or intermediate), or of a subject, or ta CSR get corrupted by an attacker. The adverse action is the modification of these object to disseminate false information on public keys and subjects and/or CAs. The result is the corruption of the whole PKI and chain of trust.

- **T.AUDIT_LOG_CORRUPTION**

  An unauthorized entity adds, corrupts, modifies or deletes audit log entries, for instance to mask their adversarial actions inside the TOE.

- **T.CONFIG_CORRUPTION**

  An unauthorized entity modifies the TOE configuration, for instance to grant them more access, or augment the TOE's attack surface.

- **T.UNAUTHENTICATED_TRANSACTIONS**

  Relying parties within an information system depend on the TOE to accurately bind subjects to their credentials for use in authenticating and providing privacy for transactions. Without the proper binding provided by the TOE, relying parties cannot ensure adequate access controls on sensitive information, ensure transactional integrity, ensure proper accountability, and/or enforce non-repudiation.

4.5 Organizational Security Policies

4.5.1 OSPs from the PP

The following organizational security policies are extracted from [PP]:

- **P.AUTHORIZED_USE_OF_INFORMATION**

  Information shall be used only for its authorized purpose.

- **P.CRYPTOGRAPHY**

  Strong cryptographic functions shall be used to perform all cryptographic operations, in accordance with [NT_CRYPTO] [RGS_B2] [RGS_B3].

  Note: this OSP was modified compared to the PP.

4.5.2 Additional OSPs

This ST additionally defines the following OSP, that has been transformed from an assumption in the PP into an OSP in the ST:

- **P.CREDENTIALS_POLICY**

  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, complexity histories, variations, etc.) Note: this policy is not applicable to biometric authentication data.

  From PP: was assumption *A.Authentication Data Management*, transformed into an OSP.

5 Security Objectives

The [PP] to which this ST refers specifies three classes of security objectives:

- security objectives that are mandatory for the TOE,
- security objectives that are mandatory for the environment,
- security objectives that must be present in the ST, but that can be fulfilled by the TOE or its environment alike.

This ST generally respects these constraints. It however places some objectives for the environment into objectives for the TOE and adds complementary security objectives.

If the text of a security objective is altered, additions are depicted in italic font, and a note is added to the description of the objective.

5.1 Security Objectives for the TOE

5.1.1 OTs from the PP

The following security objectives for the TOE (OT) are draw from TOE objectives in section 4.1 of [PP], and completed with some "optional" objectives in section 4.3 of [PP].

Note: not all OTs have been included, and some objectives have been renamed.

- **OT.CERTIFICATES**

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

  From PP: *O.Certificates*

- **OT.NON_REPUDIATION**

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

  From PP: *O.Non-repudiation*

- **OT.COMMMUNICATION_PROTECTIONS**

  Control (e.g., reroute or discard) communication traffic from an unknown source to prevent potential damage, *and protect the data in transit in all communication channels in- and out-going of the TOE (by providing integrity, authenticity and/or confidentiality of data).*

  In particular, provide a trusted path between the user (e.g., the frontend module) and the system. Provide a trusted path to security-relevant (TSF) data in which both end points have assured identities.

  From PP: consists in *O.Control unknown source communication traffic, O.Trusted path*, and additional requirements for standard integrity, authenticity and confidentiality of

data in communication channels. The rationale is that this objective deals with all communication channels in- and out-going of the TOE: from the network, and from the user (which in this case is considered as the frontend module)..

- **OT.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.

  Note: for this ST, this objective is restricted to actual import/export from the TOE, i.e. its goal is to ensure that data, when exported from the TOE, is sufficiently protected (e.g. encrypted with authenticated encryption) to be stored out of the TOE; and equivalently for the import. This objective does not cover the protection of the communication channel itself.

  From PP: *O.Data import/export*

- **OT.RESTRICT_ACTIONS_BEFORE_AUTHENTIFICATION**

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

  From PP: *O.Restrict actions before authentication*

- **OT.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.

  From PP: *O.Maintain user attributes*

- **OT.ADMIN_ACCESS_CONTROL**

  Design administrative functions so that Administrators, Operators, and Auditors only have access to the adequate functionalities corresponding to their roles.

  From PP: *O.Limitation of administrative access*

- **OT.MANAGE_SECURITY_FUNCTIONS**

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

  From PP: *O.Manage behavior of security functions*

- **OT.AUDIT**

  The TSF is capable of generating logs, and traces each action and the entity responsible for it, and can present these traces to Auditor authenticated users.

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

  From PP: *O.Individual accountability and audit records*

- **OT.PROTECT_AUDIT_LOG**

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

  From PP: *O.Protect stored audit records*

- **OT.TOE_CRYPTOGRAPHY**

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

  Note: this objective has been modified to replace FIPS-validated cryptography by ANSSI-compliant cryptography.

  From PP: *O.Cryptographic functions*, which was split in an OT and an OE.

5.1.2 Additional OTs

The following additional OTs are introduced by this ST:

- **OT.RESIDUAL_MEMORY_CLEARING**

  The TOE ensures that any sensitive data contained in a protected resource is not available when the resource is reallocated.

- **OT.SESSION_MANAGEMENT**

  The TOE provides mechanisms that mitigate the risk of unattended sessions being hijacked (session lock, session expiration, etc.). Availability of the TOE can be impacted by such mechanisms, but priority is set to prevent hijacked session.

- **OT.CA_MANAGEMENT**

  The TSF allows authorized users to create, delete, renew, modify CAs (self-signed, self-issued, cross-signed).

- **OT.CERTIFICATION**

  The TOE allows authorized users to create, delete, renew, and modify certificates.

- **OT.CRL_MANAGEMENT**

  The TSF allows to generate and export CRLs, and ensures CRLs and certificate revocation statuses are valid.

- **OT.REPLAY_PROTECTION**

  The TOE shall detect the replay of request and commands it receives from authorized users.

5.2 Security Objectives for the Environment

5.2.1 OEs from the PP

The following security objectives for the environment (OE) are draw from environment objectives in section 4.2 of [PP], and from optional OEs in section 4.3 [PP]

Note: not all OEs from the PP have been included, some have been renamed, and a few were transferred to OTs.

- **OE.ADMINISTRATOR_DOCUMENTATION**

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

  From PP: *O.Administrators, Operators, Officers and Auditors guidance documentation*

- **OE.AUDITORS_REVIEW_LOGS**

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

  From PP: *O.Auditors Review Audit Logs*

- **OE.CREDENTIALS_POLICY**

  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.

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

  From PP: merge of *O.Authentication Data Management* and *O.Disposal of Authentication Data*

- **OE.PHYSICAL_COMMUNICATION_PROTECTIONS**

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

  Note: this objective only covers the availability of the physical communication medium; it does not cover integrity, authenticity or confidentiality of the data in transfer.

  From PP: *O.Communications Protections*, restricted to the physical medium

- **OE.COMPETENT_OPERATORS**

  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.

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.

From PP: merge of *O.Competent Administrators, Operators, Officers and Auditors*, *O.Cooperative Users* (since in this ST users and operators are the same entities), and *O.CPS*.

- **OE.ENVIRONMENT_CRYPTOGRAPHY**

  The *environment* must implement *and use* approved cryptographic algorithms for encryption/decryption, authentication, and signature generation/verification; approved key generation techniques and use validated cryptographic modules. *(Validated is defined as ANSSI validated)*

  Note: similarly to P.CRYPTOGRAPHY OSP, this OE has been modified to replace FIPS-validated cryptography to ANSSI-compliant cryptography.

  From PP: *O.Cryptographic functions*¸which was split into an OT and an OE.

- **OE.SANE_INSTALL**

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

  From PP*: O.Installation*

- **OE.LIFECYCLE_SECURITY**

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

  From PP*: O.Lifecycle security*

- **OE.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.

  From PP: *O.Malicious Code Not Signed*

- **OE.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.

  From PP: *O.Notify Authorities of Security Issues*

- **OE.OPERATING_SYSTEM**

  The operating system used is validated to provide adequate security, including domain separation and non-bypassability, in accordance with the best security practices.

  In particular, the hardware platform (PC or server) and OS on which the TOE is installed is supposed up-to-date, and free of malicious code at the moment of install.

Note: removal of the mention of NIST security practices, and second paragraph added to the OE.

From PP: *O.Operating System*

- **OE.PERIODIC_INTEGRITY_CHECK**

  Provide periodic integrity checks on both system and software.

  From PP: *O.Periodically check integrity*

- **OE.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. *This includes the hardware on which the TOE runs, as well as the critical TOE dependencies such as the HSM and the database.*

  From PP: *O.Physical Protection.*

- **OE.REPAIR_SECURITY_FLAWS**

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

  From PP: *O.Repair identified security flaws.*

- **OE.SECURITY_ROLES**

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

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

  From PP: merge of *O.Security roles* and *O.User authorization management*

- **OE.SOCIAL_ENGINEERING_TRAINING**

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

  From PP: *O.Social Engineering Training.*

- **OE.BACKUP_STORAGE**

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

  From PP: *O.Sufficient backup storage and effective restoration*

- **OE.VALIDATION_OF_SECURITY_FUNCTIONS**

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

  From PP: *O.Validation of security functions.*

- **OE.PREVENT_MALICIOUS_CODE**

  Incorporate malicious code prevention procedures and mechanisms

  From PP: *O.Procedures for preventing malicious code*

- **OE.RECOVERY_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.

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

  From PP: *O.Object and data recovery free from malicious code* and *O.Preservation/trusted recovery of secure state*

- **OE.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.

  From PP: *O.Configuration Management*

- **OE.REQUIRE_INSPECTION_DOWNLOAD**

  Require inspection of downloads/transfers.

  From PP: *O.Require inspection for downloads*

- **OE.TIMESTAMPS**

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

  From PP: *O.Timestamps*

- **OE.DATA_INTEGRITY**

  Provide appropriate integrity protection to detect modifications to user and TOE data.

  From PP: data part of *O.Integrity protection of user data and software* and *O.Detect modifications of firmware, software, and backup data*

- **OE.PROTECT_DATA_INTERNAL_TRANSFER**

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

  From PP: *O.Protect user and TSF data during internal transfer*

5.2.2 Additional OEs

This ST also defines the following additional objectives for the environment:

- **OE.SECURE_KEY_STORAGE_AND_OPERATIONS**

  Private keys of CAs are securely stored, in a way that protects their confidentiality and integrity. These keys are not manipulated by the TOE, and private-key related operations are performed outside the TOE.

- **OE.AUTHENTICATE_OPERATORS**

  The environment is responsible for authenticating and verifying the identity of operators and auditors. The environment does so based on credentials it securely stores.

- **OE.CERT_REPOSITORY**

  The Operational Environment provides a certificate repository for storage of certificates, CSRs and CRLs issued by the TSF.

- **OE.KEY_ARCHIVAL**

  The Operational Environment provides the ability to use split knowledge procedures to enforce two-party control to export keys necessary to resume CA functionality if the TSF should fail.

5.3 Tracing to Threats, Policies and Assumptions

This section provides the tracing from security objectives to threats, assumptions, and organizational security policies. It also provides a rationale whenever necessary.

To be accurate, this section provides a matrix (Table 1) with the tracing. Since most of the rationale and arguments are the same as specified in the [PP], only the rationale for additional objectives, threats, assumptions or policies are provided. It was however verified that the PP's arguments are relevant in the context of this ST.

5.3.1 Tracing

In Table 1, a cell with an "X" marks a relation between the OT/OE and the Threat/Policy/Assumption.

For a relation already given in [PP] (section 8.1), a black X is used, for a relation added by this ST, the X is red; and for a relation *removed* by this ST, a red crossed-out X is used.

IDEMIA CA v1.3.1
Public Security Target

	T.CRITICAL_COMPONENT	T.FLAWED_CODE	T.MALICIOUS_CODE_EXPL	T.MESSAGE_CONTENT_TA	T.DISCLOSURE_OF_SECR	T.MODIFICATION_OF_SE	T.SENDER_DENIABILITY	T.UNAUTHORIZED_ENTIT	T.PHYSICAL_UNAUTHORI	T.WEAK_CRYPTOGRAPHY	T.RESIDUAL_MEMORY_AC	T.UNDETECTED_ACTIONS	T.CERTIFICATE_CORRUPT	T.AUDIT_LOG_CORRUPTI	T.CONFIG_CORRUPTION	T.UNAUTHENTICATED_TR	P.AUTHORIZED_USE_OF_	P.CRYPTOGRAPHY	P.CREDENTIALS_POLICY	A.AUDITORS_REVIEW_LO	A.COMPETENT_OPERATO	A.DISPOSAL_OF_AUTHEN	A.NOTIFY_AUTHORITIES_	A.SOCIAL_ENGINEERING_	A.OPERATING_SYSTEM	A.PHYSICAL_COMMUNICA	A.PHYSICAL_PROTECTION	A.SANE_INSTALL	A.HSM_SECURITY	A.TRUSTED_AUTH_SERVE	A.DATABASE_SECURITY_
OT.CERTIFICATES													X																		
OT.NON_REPUDIATION							X																								
OT.COMMMUNICATION_PROTECTIONS				X		X		X								X															
OT.DATA_IMPORT_EXPORT				X									X	X	X																
OT.RESTRICT_ACTIONS_BEFORE_AUTHENTIFIC								X								X	X														
OT.MAINTAIN_USER_ATTRIBUTES																	X														
OT.ADMIN_ACCESS_CONTROL					X			X																							
OT.MANAGE_SECURITY_FUNCTIONS	X	X										X																			
OT.AUDIT								X				X																			
OT.PROTECT_AUDIT_LOG						X								X																	
OT.TOE_CRYPTOGRAPHY				X			X	X		X						X		X													
OT.RESIDUAL_MEMORY_CLEARING											X																				
OT.SESSION_MANAGEMENT								X								X															
OT.CA_MANAGEMENT													X																		
OT.CERTIFICATION													X																		
OT.CRL_MANAGEMENT													X																		
OT.REPLAY_PROTECTION								X								X															
OE.ADMINISTRATOR_DOCUMENTATION					X																										
OE.AUDITORS_REVIEW_LOGS												X		X			X			X											
OE.CREDENTIALS_POLICY								X											X		X	X									
OE.PHYSICAL_COMMUNICATION_PROTECTIONS									X																		X				
OE.COMPETENT_OPERATORS																					X										
OE.ENVIRONMENT_CRYPTOGRAPHY				X	X	X				X			X					X													
OE.SANE_INSTALL	X		X		X																X							X			
OE.LIFECYCLE_SECURITY	X		X																												
OE.MALICIOUS_CODE_NOT_SIGNED			X																												

	T.CRITICAL_COMPONENT	T.FLAWED_CODE	T.MALICIOUS_CODE_EXPL	T.MESSAGE_CONTENT_TA	T.DISCLOSURE_OF_SECR	T.MODIFICATION_OF_SE	T.SENDER_DENIABILITY	T.UNAUTHORIZED_ENTIT	T.PHYSICAL_UNAUTHORI	T.WEAK_CRYPTOGRAPHY	T.RESIDUAL_MEMORY_AC	T.UNDETECTED_ACTIONS	T.CERTIFICATE_CORRUPT	T.AUDIT_LOG_CORRUPTI	T.CONFIG_CORRUPTION	T.UNAUTHENTICATED_TR	P.AUTHORIZED_USE_OF_	P.CRYPTOGRAPHY	P.CREDENTIALS_POLICY	A.AUDITORS_REVIEW_LO	A.COMPETENT_OPERATO	A.DISPOSAL_OF_AUTHEN	A.NOTIFY_AUTHORITIES_	A.SOCIAL_ENGINEERING_	A.OPERATING_SYSTEM	A.PHYSICAL_COMMUNICA	A.PHYSICAL_PROTECTION	A.SANE_INSTALL	A.HSM_SECURITY	A.TRUSTED_AUTH_SERVE	A.DATABASE_SECURITY_
OE.PREVENT_MALICIOUS_CODE			X																												
OE.NOTIFY_AUTHORITIES_OF_SECURITY_ISSU								X															X								
OE.OPERATING_SYSTEM			X	X							X			X	X										X						
OE.PERIODIC_INTEGRITY_CHECK			X									X																			X
OE.PHYSICAL_PROTECTION									X																			X		X		X
OE.REPAIR_SECURITY_FLAWS	X	X																													
OE.SECURITY_ROLES													X				X													X	
OE.SOCIAL_ENGINEERING_TRAINING					X																			X							
OE.BACKUP_STORAGE	X																														X
OE.VALIDATION_OF_SECURITY_FUNCTIONS			X																												
OE.RECOVERY_FROM_MALICIOUS_CODE	X		X			~~X~~																									
OE.CONFIGURATION_MANAGEMENT	X		X												X																
OE.REQUIRE_INSPECTION_DOWNLOAD			X																												
OE.TIMESTAMPS	X											X	X	X																	
OE.DATA_INTEGRITY						X							X	X	X																
OE.PROTECT_DATA_INTERNAL_TRANSFER				X	X								X	X	X																
OE.SECURE_KEY_STORAGE_AND_OPERATIONS					X	X																							X		
OE.AUTHENTICATE_OPERATORS								X								X														X	
OE.CERT_REPOSITORY													X																		
OE.KEY_ARCHIVAL					X	X																							X	X	

Table 1 - Tracing from security objectives to threats/assumptions/policies

5.3.2 Rationale

The rationale provides justifications for the tracing between Threats/OSPs/Assumptions to Security Objectives.

It borrows a large part of the rationale from [PP]. More exactly, this section addresses all the differences with the tracing from [PP], and thus completes the rationale.

Deleted SPD elements

The following Threats, OSPs, Assumption, and Objectives are in the [PP], and are not included in this ST:

- Threats:
  - *T.Administrative errors of omission;*
  - *T.Administrators, Operators, Officers and Auditors commit errors or hostile actions;*
  - *T.User abuses authorization to collect and/or send data;*
  - *T.User error makes data inaccessible;*
  - *T.Social Engineering;*
- Objectives :
  - *O.Preservation/trusted recovery of secure state* was transferred from OT to OE (because the PP does not trace any SFR to that OT);
  - "software part" of *O.Integrity protection of user data and software* and *O.Detect modifications of firmware, software, and backup data*;
  - *OT.React to detected attacks*;
  - *OT.Respond to possible loss of stored audit records;*
- Assumption:
  - *A.Malicious Code Not Signed.*

OT.CERTIFICATES

[Added] Counters T.CERTIFICATE_CORRUPTION by ensuring that certificates and CRLs, and the information they contain are valid.

OT.COMMMUNICATION_PROTECTIONS

Note: this objective is *O.Control unknown source communication traffic* along with *O.Trusted Path* from [PP], augmented with traditional protection for data in transit.

[Added] Counters T.MESSAGE_CONTENT_TAMPERING by protecting data in transit in communication channels.

[Added] Counters T.MODIFICATION_OF_SECRET_KEYS since modifications of secret keys also happen during communications between components. Indeed, the frontend component take in the HSM password from a CA Operator and sends it to the IDEMIA CA component. The frontend and IDEMIA CA components may be on different physical machines.

[Added] Counters T.UNAUTHENTICATED_TRANSACTIONS since this threat is close to T.UNAUTHORIZED_ENTITY_ACCESS, which is countered by the objective.

Note : the objective counters T.UNAUTHORIZED_ENTITY_ACCESS for two reasons: because the PP states that *O.Control unknown source communication traffic* counters *T.Hacker gains access* (by verifying the source of received data), and because T.UNAUTHORIZED_ENTITY_ACCESS contains the notion of credential theft (which are communicated across the network in secure channels).

OT.DATA_IMPORT_EXPORT

Note: this objective was slightly changed to cover only the case of actual data import/export of the TOE. Protection of communications is the object of the above objective.

[Added] Counters T.CERTIFICATE_CORRUPTION, T.AUDIT_LOG_CORRUPTION, and T.CONFIG_CORRUPTION, since certificates/CRLs/CSRs, audit logs and configurations are the three objects that can be imported and/or exported from the TOE.

OT.RESTRICT_ACTIONS_BEFORE_AUTHENTIFICATION

[Added] Counters T.UNAUTHENTICATED_TRANSACTIONS since this threat is close to T.UNAUTHORIZED_ENTITY_ACCESS, which is countered by the objective.

OT.ADMIN_ACCESS_CONTROL

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS, since this OT ensures that only the authorized entity access the parts of the TOE they are allowed to.

OT.MANAGE_SECURITY_FUNCTIONS

[Added] Counters T.FLAWED_CODE by requiring operational security during the development of the application.

[Added] Counters T.UNDETECTED_ACTIONS by providing the audit log functionality so that Auditors can detect actions.

OT.AUDIT

[Added] Counters T.UNDETECTED_ACTIONS by requiring that each action on the TOE be included in the audit log.

OT.PROTECT_AUDIT_LOG

[Added] Counters T.AUDIT_LOG_CORRUPTION by requiring that the log be protected against unauthorized access, modification or deletion.

OT.TOE_CRYPTOGRAPHY

Note: this OT is a part of an OE from the PP that was transformed into an OT in this ST; and the requirement was changed from FIPS-validated cryptography to ANSSI-validated cryptography.

[Added] Directly counters T.WEAK_CRYPTOGRAPHY by requiring ANSSI-validated, strong cryptography.

[Added] Counters T.MESSAGE_CONTENT_TAMPERING and T.SENDER_DENIABILITY, since the TOE uses cryptography to verify the integrity and authenticity of messages. Using strong cryptography thus participates in preventing the threats.

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS and T.UNAUTHENTICATED_TRANSACTIONS since cryptography is used by the TOE to authenticate

users (to verify the access token). Using strong cryptography thus participates in preventing those threats.

OT.RESIDUAL_MEMORY_CLEARING (new OT)

[Added] Directly counters T.RESIDUAL_MEMORY_ACCESS by ensuring that the TOE deletes sensitive data after use in memory.

OT.SESSION_MANAGEMENT (new OT)

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS and T.UNAUTHENTICATED_TRANSACTIONS by automatically managing the operators' sessions (by closing it, locking it, or having it expire). This prevents malicious reuse of unattended or open sessions by unauthorized entities.

OT.CA_MANAGEMENT (new OT)

[Added] Counters T.CERTIFICATE_CORRUPTION by allowing operators to manage CA certificates, and put in place a PKI. In particular, it allows administrators to revoke, delete or modify CA certificates for keys that were compromised.

OT.CERTIFICATION (new OT)

[Added] Counters T.CERTIFICATE_CORRUPTION by allowing operators to manage end-entities certificates, in particular by using cryptographic means to protect the contents of certificates..

OT.CRL_MANAGEMENT (new OT)

[Added] Counters T.CERTIFICATE_CORRUPTION by allowing administrators to manage and export CRLs. In particular, it allows administrators to update the revocation status of certificates, for keys that were compromised, and communicate updated CRLs.

OT.REPLAY_PROTECTION (new OT)

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS and T.UNAUTHENTICATED_TRANSACTIONS by preventing the replay of legitimate operators' messages and commands to the TOE.

OE.AUDITORS_REVIEW_LOGS

[Added] Counters T.UNDETECTED_ACTIONS, since frequently reviewing audit log participates in ensuring no important security event remains undetected in the system.

[Added] Counters T.AUDIT_LOG_CORRUPTION since frequently reviewing audit log helps towards detecting unauthorized modifications to the audit log.

OE.CREDENTIALS_POLICY

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS in the sense that a robust credential policy (strong passwords, periodic password changes, deactivation of obsolete accounts and privileges, …) participates in preventing and mitigating credential theft, and therefore unauthorized access to the TOE by an attacker.

[Added] Supports OSP P.CREDENTIALS_POLICY: this policy was noted as an assumption in [PP], but the argumentation is the same.

OE.PHYSICAL_COMMUNICATION_PROTECTIONS

[Added] Counters T.PHYSICAL_UNAUTHORIZED_ENTITY_ACCESS, since it protects physical access to the TOE through the physical communication mediums connected to the TOE or TOE's environment.

OE.ENVIRONMENT_CRYPTOGRAPHY

[Added] Counters T.MESSAGE_CONTENT_TAMPERING, since the TOE's environment uses cryptography (notably within the TLS protocol) to verify the integrity and authenticity of messages. Using strong cryptography thus participates in preventing the threat.

[Added] Directly counters T.WEAK_CRYPTOGRAPHY, by requiring strong and approved cryptographic mechanisms to be used in the TOE's environment.

[Added] Counters T.CERTIFICATE_CORRUPTION, since the environment protects against this threats using cryptography (inside the HSM) to digitally sign the certificates. Employing robust cryptographic mechanisms thus participates in countering these threats.

OE.SANE_INSTALL

[Added] Is directly supported by A.SANE_INSTALL (this assumption was added in the ST).

[Added] Counters T.MALICIOUS_CODE_EXPLOITATION, in the sense that a sane installation by competent administrators reduces the risk for an attacker successfully introducing or executing malicious code in the TOE.

[Added] Counters T.DISCLOSURE_OF_SECRET_KEYS and T.MODIFICATION_OF_SECRET_KEYS, since the installation involves the setup of the HSM with the competent administrators, as well as the initial configuration of the TOE (and in particular the cryptographic material used by the TLS channels of the TOE).

OE.MALICIOUS_CODE_NOT_SIGNED

[Added] Counters T.MALICIOUS_CODE_EXPLOITATION, trivially.

OE.OPERATING_SYSTEM

[Added] Can participate in countering T.MALICIOUS_CODE_EXPLOITATION, if the OS have some mechanisms ensuring that non signed code can not run on the machine.

[Added] Counters T.MESSAGE_CONTENT_TAMPERING, for *local* messages exchanged between the TOE and the OS (i.e. messages within the local system), by providing secure inter-process or process-kernel communication APIs.

[Added] Can participate in countering T.RESIDUAL_MEMORY_ACCESS, since some OSes erase (re)allocated pages before handing them to processes. This acts as a second protection, in addition to the memory erasures performed in the TOE.

[Added] Counters T.AUDIT_LOG_CORRUPTION and T.CONFIG_CORRUPTION by enforcing access control mechanisms on files associated with the audit log and the configuration of the TOE.

OE.PERIODIC_INTEGRITY_CHECK

[Added] Counters T.UNDETECTED_ACTIONS since it allows the detection of software modifications through periodic software integrity checks.

[Added] Upheld by A.DATABASE_SECURITY_AND_BACKUP, since this assumption states that the integrity of the database used by the TOE is ensured by the environment.

OE.PHYSICAL_PROTECTION

[Added] Upheld by A.HSM_SECURITY and A.DATABASE_SECURITY_AND_BACKUP, since these two assumptions state that the HSM and the hardware on which the database run are both protected against physical tampering.

OE.SECURITY_ROLES

[Added] Upheld by A.TRUSTED_AUTH_SERVER since the authentication server is responsible for storing the roles of users, and providing the information about these roles to the TOE inside the access tokens.

[Added] Counters T.CERTIFICATE_CORRUPTION since the modification of certificates is restricted to the CA Operator role.

OE.SOCIAL_ENGINEERING_TRAINING

[Added] Counters T.DISCLOSURE_OF_SECRET_KEYS since an attacker can possibly gain knowledge of secret or private keys through social engineering, i.e. by convincing of tricking administrators to export and leak secret keys.

OE.BACKUP_STORAGE

[Added] Is supported by A.DATABASE_SECURITY_AND_BACKUP, since this assumption states that the database used by the TOE puts in places the best practices in terms of data backup.

OE.RECOVERY_FROM_MALICIOUS_CODE

[Removed] No longer counters T.MODIFICATION_OF_SECRET_KEYS, since the TOE never directly manipulates or even learns the value of private keys (they are managed and stay in the HSM).

OE.CONFIGURATION_MANAGEMENT

[Added] Counters T.CONFIG_CORRUPTION and T.AUDIT_LOG_CORRUPTION, since the objective involves controlling changes to configuration items. This mitigates errors of omissions, and allows to detect malicious corruptions.

OE.TIMESTAMPS

[Added] Counters T.UNDETECTED_ACTIONS and T.AUDIT_LOG_CORRUPTION in the sense that the audit log entries rely on reliable timestamping in order to record the sequencing of events.

[Added] Counters T.CERTIFICATE_CORRUPTION in the sense that the provision of reliable timestamps by the environment ensures that the validity dates included in certificates will be correct.

OE.DATA_INTEGRITY

[Added] Counters T.CERTIFICATE_CORRUPTION, T.AUDIT_LOG_CORRUPTION, and T.CONFIG_CORRUPTION, by requiring that these objects (in particular) must be protected in integrity while stored in the environment (in database, on the disk, …).

OE.PROTECT_DATA_INTERNAL_TRANSFER

[Added] Counters T.CERTIFICATE_CORRUPTION, T.AUDIT_LOG_CORRUPTION, and T.CONFIG_CORRUPTION, since it protects these assets (certificates/CRLs/CSRs, audit logs and

configurations) while they are in transit in the system and between the TOE and its environment.

OE.SECURE_KEY_STORAGE_AND_OPERATIONS (new OE)

[Added] Counters T.DISCLOSURE_OF_SECRET_KEYS and T.MODIFICATION_OF_SECRET_KEYS since this objective for the environment is precisely about protecting CA private keys (in the HSM).

[Added] Upheld by A.HSM_SECURITY, since it is the HSM that is responsible for the secure storage of CA private keys, and for performing CA private-key related operations.

OE.AUTHENTICATE_OPERATORS (new OE)

[Added] Counters T.UNAUTHORIZED_ENTITY_ACCESS in the sense that this objective requires that credentials are securely stored by the operation environment.

[Added] Upheld by A.TRUSTED_AUTH_SERVER, since it is the authentication server, part of the TOE's environment) that stores the credentials, and is responsible for verifying the user provided credentials against the ones that are stored.

OE.CERT_REPOSITORY (new OE)

[Added] Counters T.CERTIFICATE_CORRUPTION in the sense that the database used by the TOE is responsible for ensuring the integrity of certificates/CRLs/CSRs that it stores.

[Added] Is supported by A.DATABASE_SECURITY_AND_BACKUP, since this assumption states that the database used by the TOE is secure, and in particular ensures the integrity of the objects it stores.

OE.KEY_ARCHIVAL (new OE)

[Added] Counters T.MODIFICATION_OF_SECRET_KEYS and T.DISCLOSURE_OF_SECRET_KEYS, since it ensures that no *single* party can extract, export or import CA private keys into the HSM.

[Added] Upheld by A.HSM_SECURITY, since it is the HSM that is responsible for this security feature.

6 Extended Requirements

This ST claims to be CC Part 3 Conformant. There is no extended SAR component.

This ST claims to be CC Part 2 Extended. It uses two kinds of extended functional requirements (SFRs):

- those defined by the protection profile [PP]. They are noticeable by the presence of the suffix "_CIMC" in their name, e.g. FCO_NRO_CIMC.4. The definition of these SFRs are not reproduced here, and the reader may refer to the public [PP].
- other SFRs that are defined below. The reason for each extended SFR is given appropriately.

6.1 SFR FMT_MOF_IDA.3

Below is the definition of the SFR FMT_MODF_IDA.3, which is a version slightly adapted from FMT_MOF_CIMC.3 in the protection profile [PP]. This SFR deals with Certificate Profile Management.

6.1.1 Rationale

The redefinition of the SFR is necessary, mainly because the version from the protection profile *requires* that operators define the acceptable values for some X509 certificate extensions. Yet, the TOE only *allows* operator to define acceptable values for X509 extension, but does not *force* them to do so. The main change thus lies in the substitution of the phrase "shall require" into "shall allow" in FMT_MOF_CIMC.3.3. The redefinition also makes minor changes, adapting the SFR to a scenario that sticks closer to the TOE (these changes can be considered as a *refinement* operation, and only make the SFR more constraining).

Changes from FMT_MOF_CIMC.3 are underlined in the definition below.

6.1.2 Definition

FMT_MOF_IDA.3 EXTENDED CERTIFICATE PROFILE MANAGEMENT

FMT_MOF_IDA.3.1 The TSF shall implement certificate profiles and shall ensure that issued certificates are consistent with at least one of the defined profiles.

FMT_MOF_IDA.3.2 The TSF shall require the CA Operator, RA Operator or Integrating Application to specify the set of acceptable values for the following fields:

- the key owner's identifier;
- the signature algorithm identifier for the certificate signature;
- the identifier of the certificate issuer;
- the length of time for which the certificate is valid.

FMT_MOF_IDA.3.3 If the certificates generated are X.509 public key certificates, the TSF shall allow the CA Operator, RA Operator or Integrating Application to specify or select a profile with the set of acceptable values to any of the following extensions:

- authority key identifier;
- subject key identifier;
- key usage;
- private key usage period;
- certificate policies;
- policy mappings;
- subject alternative name;
- issuer alternative name;
- subject directory attribute;
- basic constraints;
- name constraints;
- policy constraints;
- extended key usage;
- CRL distribution points;
- inhibit any policy;
- authority information access;
- subject information access;
- name change;
- document type list;
- QC statement;
- biometric information.

FMT_MOF_IDA.3.4 The CA Operator, RA Operator or Integrating Application shall specify the acceptable set of certificate extensions for each defined certificate profile.

6.1.3 Dependencies

The dependencies of FMT_MOF_IDA.3 are the same as FMT_MOF_CIMC.3 in the protection profile [PP]:

- FMT_MOF.1
- FMT_SMR.1

7 Security Requirements

7.1 Security Functional Requirements (SFR)

7.1.1 Definitions of Subjects, Objects, Operations and Security Attributes

This section defines all terms used in the SFRs: objects, subjects, security attributes, operations, SFPs, External Entities. Lastly, it defines the notations used for performing the SFR operations (assignment, selection, etc.).

7.1.1.1 Users and Roles

The PP however uses the roles of *Administrators, Officers, Operators* and *Auditors.,* while the ST uses *System Administrator, CA Operator, RA Operator, Auditor* and *Integrating Application.* The correspondence from the user roles in the ST to user roles in the PP is given in section 4.1.

7.1.1.2 Subjects

Note that *subjects,* in the CC reference, are not users, but entities *within* the TOE that perform the action. Typically, a process or thread.

7.1.1.3 Objects

The objects that appear in the functional requirement are the assets described in section 4.2.

7.1.1.4 Security Attributes

The security attributes of users/subjects are:

- their *role* (included in the access token),
- their *username* (included in the access token),
- an *access token validity date*,
- an *access token signature*,

The security attributes of objects are:

- **DT.CA_CERT**, **DT.END_ENTITY_CERT** and **DT.END_ENTITY_CSR**
- **DT.CA_CRL**

7.1.1.5 Operations

The TSF involves the following *operations* among subjects and objects:

- **OP.READ_CONFIG**
- **OP.READ_TLS_KEYS**
- **OP.FWD_HTTP_REQS**
- **OP.VERIF_AUTH_TOKEN**
- **OP.PRIVILEGED_WRITE_CA_AND_CERTS**
- **OP.WRITE_CA_AND_CERTS**
- **OP.WRITE_CSR**
- **OP.READ_CERTS**
- **OP.WRITE_AUDIT_LOG**
- **OP.READ_AUDIT_LOGS**
- **OP.AUTHORIZE_HSM**
- **OP.UNAUTHORIZE_HSM**
- **OP.FWD_PRIVKEY_REQS**
- **OP.USE_PRIVKEY**

7.1.1.6 External entities

The external entities are:

- the **Authentication server**, which produces and signs the user authentication tokens
- the **Frontend component**, which takes user input and forwards them to the TOE.

7.1.1.7 SFP Names

The Security Function Policies defined and used in this ST are names as follows:

- **SFP.TOE_ACCESS_CONTROL_POLICY**

7.1.1.8 Operations on SFRs: Notations

The operations on SFR are as follows:

- **Assignment**: underlined text
- **Selection**: underlined text (the difference with assignment operations is made by looking at the SFR definition)

- **Refinement**: underlined text (the difference with assignment and selection operations is made by looking at the SFR definition)

- **Iteration**: by repeating the SFR and using the characters "<" and ">" to name the iteration. Example CLASS_COMP.1.<ITERATION_1_NAME>. For iterated SFRs which require assignments/selections, a table is also provided. Each line is an iteration, the first column indicating the name of the iteration, the second column indicating the first assignment/selection, the third column the second assignment/selection, etc.

7.1.2 SFRs from the PP

This section describes the SFRs considered for this ST. To be consistent with the [PP], the subsections are not organized by SFR classes, but by logical security functionality.

The full text of SFRs is reproduced in the ST.

7.1.2.1 Security Audit

FAU_GEN.1 AUDIT DATA GENERATION

FAU_GEN.1.1 The TSF shall be able to generate an audit record.

FAU_GEN.1.2 The TSF shall record within each audit record.

FAU_GEN.2 USER IDENTITY ASSOCIATION

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

FAU_SEL.1 SELECTIVE AUDIT

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events.

7.1.2.2 Roles

FMT_MOF.1 MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOR

FMT_MOF.1.1 The TSF shall restrict the ability to modify the behavior of functions to the authorized roles.

7.1.2.3 Access Control

FDP_ACC.1 SUBSET ACCESS CONTROL

FDP_ACC.1.1 The TSF shall enforce SFP.TOE_ACCESS_CONTROL_POLICY.

FDP_ACF.1 SECURITY ATTRIBUTE BASED ACCESS CONTROL

FDP_ACF.1.1 The TSF shall enforce SFP.TOE_ACCESS_CONTROL_POLICY to objects.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the user role.

7.1.2.4 Identification and Authentication

FIA_UAU.1 TIMING OF AUTHENTICATION

FIA_UAU.1.1 The TSF shall allow no actions on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UID.1 TIMING OF IDENTIFICATION

FIA_UID.1.1 The TSF shall allow no actions on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 USER-SUBJECT BINDING

FIA_USB.1.1 The TSF shall associate user security attributes with subject acting on behalf of that user.

FIA_USB.1.2 The TSF shall enforce rules on the initial association of user security attributes with subjects acting on the behalf of users.

FIA_USB.1.3 The TSF shall enforce rules governing changes to the user security attributes associated with subjects acting on the behalf of users.

7.1.2.5 Remote Data Entry and Export

FCO_NRO_CIMC.3 ENFORCED PROOF OF ORIGIN AND VERIFICATION OF ORIGIN

The ST makes several iterations of this SFR, for the different objects that have proofs of origins.

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 see the iteration table below 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.

SFR Iteration	Assignment
FCO_NRO_CIMC.3<CSR>	Public key of **DT.END_ENTITY_CSR** of the CA or End-Entity.
FCO_NRO_CIMC.3<CERTIFICATE>	Public key of **DT.CA_CERT** or **DT.END_ENTITY_CERT** belonging to the CA or End entity
FCO_NRO_CIMC.3<CRL>	Public key of **DT.CA_CRL**
FCO_NRO_CIMC.3<AUTH_TOKEN>	Public key of the **Authentication server (external entity)**

FDP_UCT.1 BASIC DATA EXCHANGE CONFIDENTIALITY

FDP_UCT.1.1 The TSF shall enforce **SFP.TOE_ACCESS_CONTROL_POLICY** to transmit user data in a manner protected from unauthorized disclosure.

FPT_ITC.1 INTER-TSF CONFIDENTIALITY DURING TRANSMISSION

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission.

FCO_NRO_CIMC.4 ADVANCED VERIFICATION OF ORIGIN

FCO_NRO_CIMC.4.1 The TSF shall, for initial certificate registration messages sent by the certificate subject, only accept messages protected using an authentication code, keyed hash, or digital signature algorithm.

FCO_NRO_CIMC.4.2 The TSF shall, for all other security-relevant information, only accept the information if it was signed using a digital signature algorithm.

FDP_CIMC_CSE.1 CERTIFICATE STATUS EXPORT

FDP_CIMC_CSE.1.1 Certificate status information shall be exported from the TOE in messages whose format complies with the X.509 standard for CRLs of X509 certificates (note: they are not produced and thus not exported for CVC/EAC certificates).

7.1.2.6 Certificate Revocation List Profile Management

FMT_MOF_CIMC.5 EXTENDED CERTIFICATE REVOCATION LIST PROFILE MANAGEMENT

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 user role to specify the set of acceptable values for the following fields and extensions:

- issuer;
- issuerAltName (note: If a CIMC does not issue CRLs with this extension, then it is not required within the certificate revocation list profile.)
- nextUpdate (i.e., a promise of next CRL in specified time).

FMT_MOF_CIMC.5.3 If the TSF issues CRLs, the CA Operator, RA Operator or Integrating Application shall specify the acceptable set of CRL and CRL entry extensions.

7.1.2.7 Certificate Registration

FDP_CIMC_CER.1 CERTIFICATE GENERATION

This SFR is modified in FDP_CIMC_CER_1.4.

FDP_CIMC_CER.1.1 The TSF shall only generate certificates whose format complies with the X.509 standard for public key certificates [RFC5280] or the TR-03110 standard [BSI-TR03110] for EAC 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.

7.1.2.8 Certificate Revocation

FDP_CIMC_CRL.1 CERTIFICATE REVOCATION LIST VALIDATION

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.

7.1.2.9 Strength of Function Requirements

This section defines functional requirements for cryptography with ANSSI-compliant cryptography. The SFR is thus modified.

FCS_SOF_CIMC.1 CIMC STRENGTH OF FUNCTIONS

FCS_SOF_CIMC.1.1 The TSF shall provide cryptographic mechanisms.

All cryptographic operations performed (including key generation) *by the TOE or at the request of the TOE* shall be performed with ANSSI-compliant cryptographic algorithms and keys, as per the ANSSI's *Référentiel Général de Sécurité* (RGS).

The algorithms specified shall be performed using cryptographic algorithms and mechanisms that are compliant with ANSSI's "*Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques*" [NT_CRYPTO] for a usage *beyond* the year 2030.

Authentication procedures shall be implemented as specified by ANSSI's *Référentiel Général de Sécurité* (RGS), Appendix B3 "*Règles et recommandations concernant les mécanismes d'authentification*" [RGS_B3].

7.1.3 Additional SFR for this ST

7.1.3.1 Security Audit

FAU_SAR.1 AUDIT REVIEW

FAU_SAR.1.1 The TSF shall provide Auditors with the capability to read all audit information from the audit records.

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

FAU_SAR.2 RESTRICTED AUDIT REVIEW

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

FAU_SAR.3 SELECTABLE AUDIT REVIEW

FAU_SAR.3.1 The TSF shall provide the ability to apply the method of selection.

7.1.3.2 Roles

FIA_ATD.1 USER ATTRIBUTE DEFINITION

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: the user's name and role.

FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES (REFINED)

FMT_SMR.2.1 The TSF shall maintain user roles.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions in the following list are satisfied.

- only 1 role per user is allowed by the TOE.
- users with multiple roles are not allowed to access the TSF even if one of the user's role satisfy the access rights.

FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1 The TSF shall be capable of performing the same functions as specified in FMT_MOF.1.

FMT_MTD.1 MANAGEMENT OF TSF DATA

FMT_MTD.1.1 The TSF shall restrict the ability to create, query, modify, and/or delete, the certificates, certificate profiles, certificate requests, certificate revocation lists, CAs, audit log to the roles defined by the Access Control.

7.1.3.3 Secure Memory Erasure

FDP_RIP.1 SUBSET RESIDUAL INFORMATION PROTECTION

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource.

7.1.3.4 Session Management

FIA_UAU.6 RE-AUTHENTICATING

FIA_UAU.6.1 The TSF shall re-authenticate the user.

FTA_SSL.3 TSF-INITIATED TERMINATION

FTA_SSL.3.1 The TSF shall terminate an interactive session after the token DT.AUTH_TOKEN of the user is no longer valid.

7.1.3.5 Replay Protection

FPT_RPL.1 REPLAY DETECTION

FPT_RPL.1.1 The TSF shall detect replay.

FPT_RPL.1.2 The TSF shall perform ignore the replayed request when replay is detected.

7.1.3.6 Certificate Profile Management

FMT_MOF_IDA.3 EXTENDED CERTIFICATE PROFILE MANAGEMENT

See the definition of the SFR in Section 6 "Extended Requirements". The instantiation here is identical to the definition, since the SFR does not involve any operation.

7.1.3.7 Certificate, Certificate Requests and CRL Authenticity

FDP_DAU.1 BASIC DATA AUTHENTICATION

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **DT.CA_CERT**, **DT.CA_CRL**, **DT.END_ENTITY_CERT**, and **DT.END_ENTITY_CSR**.

FDP_DAU.1.2 The TSF shall provide anyone (i.e. any entity, component or user) with the ability to verify evidence of the validity of the indicated information.

7.1.3.8 Trusted Path

FTP_TRP.1 TRUSTED PATH

FTP_TRP.1.1 The TSF shall provide a communication path between itself and remote and/or local 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 and disclosure.

FTP_TRP.1.2 The TSF shall permit remote and/or local users to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for all API endpoints.

7.2 Security Assurance Requirements (SAR)

This chapter defines the list of the assurance measures required for the TOE security assurance requirements.

The level aimed for is EAL4, augmented with

- ALC_FLR.1

The full list of SARs is provided in the dependency analysis, section 7.3.3.

7.3 Security Requirements Rationale

7.3.1 SFR Dependencies Rationale

Table 2 lists the SFRs used in this ST. It also shows the dependencies of each SFR.

Class	Component	Dependencies
FAU	FAU_GEN.1	-
	FAU_GEN.2	FAU_GEN.1, FIA_UID.1
	FAU_SEL.1	FAU_GEN.1, FMT_MTD.1
	FAU_SAR.1	FAU_GEN.1
	FAU_SAR.2	FAU_SAR.1
	FAU_SAR.3	FAU_SAR.1
FCO	FCO_NRO_CIMC.3	FIA_UID.1
	FCO_NRO_CIMC.4	FCO_NRO_CIMC.3
FCS	FCS_SOF_CIMC.1	-
FDP	FDP_ACC.1	FDP_ACF.1
	FDP_ACF.1	FDP_ACC.1
	FDP_CIMC_CER.1	-
	FDP_CIMC_CRL.1	-
	FDP_CIMC_CSE.1	-
	FDP_DAU.1	-
	FDP_UCT.1	FDP_ACC.1 and FTP_TRP.1

Class	Component	Dependencies
	FDP_RIP.1	-
FIA	FIA_ATD.1	-
	FIA_UAU.1	FIA_UID.1
	FIA_UAU.6	-
	FIA_UID.1	-
	FIA_USB.1	FIA_ATD.1
FMT	FMT_MOF.1	FMT_SMR.1[7], FMT_SMF.1
	FMT_SMF.1	-
	FMT_SMR.2	FIA_UID.1
	FMT_MTD.1	FMT_SMR.1[7], FMT_SMF.1
	FMT_MOF_IDA.3	FMT_MOF.1, FMT_SMR.1[7]
	FMT_MOF_CIMC.5	FMT_MOF.1, FMT_SMR.1[7]
FPT	FPT_ITC.1	-
	FPT_RPL.1	-
FTA	FTA_SSL.3	-
FTP	FTP_TRP.1	-

Table 2 - Dependencies Rationale for SFRs

7.3.2 SFR to Security Objectives Rationale

This section provides the tracing from SFRs to OTs, i.e., it states which SFRs participates in meeting which OT. It also provides a rationale whenever necessary.

Since most of the rationale and arguments are the same as specified in the [PP], only the rationale for additional objectives, threats, assumptions or policies are provided. It was however verified that the PP's arguments are relevant in the context of this ST.

7.3.2.1 Tracing

In Table 3, a cell with an "X" marks a relation between an OT and a SFR (the SFR meets or participate in meeting the OT).

For a relation already given in the [PP] (section 8.2.1), a black X is used, and for a relation added by this ST, the X is red. No tracing was *removed* compared to the PP.

[7] Note that this dependency is satisfied because FMT_SMR.2, which is included in the ST, is hierarchical to FMT_SMR.1.

	OT.CERTIFICATES	OT.NON_REPUDIATION	OT.COMMUNICATION_PROTECTIONS	OT.DATA_IMPORT_EXPORT	OT.RESTRICT_ACTIONS_BEFORE_AUTHENTIFICATION	OT.MAINTAIN_USER_ATTRIBUTES	OT.ADMIN_ACCESS_CONTROL	OT.MANAGE_SECURITY_FUNCTIONS	OT.AUDIT	OT.PROTECT_AUDIT_LOG	OT.TOE_CRYPTOGRAPHY	OT.RESIDUAL_MEMORY_CLEARING	OT.SESSION_MANAGEMENT	OT.CA_MANAGEMENT	OT.CERTIFICATION	OT.CRL_MANAGEMENT	OT.REPLAY_PROTECTION
FAU_GEN.1									X								
FAU_GEN.2									X								
FAU_SEL.1									X								
FAU_SAR.1									X								
FAU_SAR.2							X		X								
FAU_SAR.3									X								
FCO_NRO_CIMC.3.<CSR>		X	X														
FCO_NRO_CIMC.3.<CERTIFICATE>		X	X														
FCO_NRO_CIMC.3.<CRL>		X	X														
FCO_NRO_CIMC.3.<AUTH_TOKEN>		X	X														
FCO_NRO_CIMC.4		X															
FCS_SOF_CIMC.1											X						
FDP_ACC.1							X										
FDP_ACF.1							X										
FDP_CIMC_CER.1	X													X	X		
FDP_CIMC_CRL.1	X													X		X	
FDP_CIMC_CSE.1	X															X	
FDP_DAU.1	X													X	X	X	
FDP_RIP.1												X					
FDP_UCT.1				X													
FIA_UAU.1					X		X										
FIA_UAU.6					X								X				
FIA_UID.1							X		X								

	OT.CERTIFICATES	OT.NON_REPUDIATION	OT.COMMUNICATION_PROTECTIONS	OT.DATA_IMPORT_EXPORT	OT.RESTRICT_ACTIONS_BEFORE_AUTHENTIFICATION	OT.MAINTAIN_USER_ATTRIBUTES	OT.ADMIN_ACCESS_CONTROL	OT.MANAGE_SECURITY_FUNCTIONS	OT.AUDIT	OT.PROTECT_AUDIT_LOG	OT.TOE_CRYPTOGRAPHY	OT.RESIDUAL_MEMORY_CLEARING	OT.SESSION_MANAGEMENT	OT.CA_MANAGEMENT	OT.CERTIFICATION	OT.CRL_MANAGEMENT	OT.REPLAY_PROTECTION
FIA_USB.1						X											
FIA_ATD.1						X											
FMT_MOF.1								X									
FMT_MTD.1							X		X	X							
FMT_SMF.1							X	X									
FMT_SMR.2						X											
FMT_MOF_IDA.3														X	X		
FMT_MOF_CIMC.5																X	
FPT_ITC.1			X	X													
FPT_RPL.1																	X
FTA_SSL.3													X				
FTP_TRP.1			X		X												

Table 3 - Tracing from SFR to OTs

7.3.3 SAR Dependencies Rationale

The following table shows the exact SAR chosen, along with their dependencies. The SAR in bold denotes the augmentation from EAL4. The last column shows the dependencies of the augmented SARs, which are all satisfied.

Class	Component	Dependences
ADV	ADV_ARC.1	-
	ADV_FSP.4	-
	ADV_IMP.1	-
	ADV_TDS.3	-
AGD	AGD_OPE.1	-
	AGD_PRE.1	-
ALC	ALC_CMC.4	-
	ALC_CMS.4	-
	ALC_DEL.1	-
	ALC_DVS.1	-
	ALC_FLR.1	None
	ALC_LCD.1	-
	ALC_TAT.1	-
ASE	ASE_CCL.1	-
	ASE_ECD.1	-
	ASE_INT.1	-
	ASE_OBJ.2	-
	ASE_REQ.2	-
	ASE_SPD.1	-
	ASE_TSS.1	-
ATE	ATE_COV.2	-
	ATE_DPT.1	-
	ATE_FUN.1	-
	ATE_IND.2	-
AVA	AVA_VAN.3	-

7.3.4 SAR Rationale

The rationale for the augmentations compared to the EAL4 package are as follows:

ALC_FLR.1

The flaw remediation assurance improves a rigorous management for updating the TOE.