CombICAO Applet in SSCD Configuration on Cosmo v9

Public Security Target

- Printed versions of this document are uncontrolled -

DOCUMENT MANAGEMENT

Business Unit – Department	**PSI**
Document type	**Public FQR**
Document Title	**CombICAO Applet in SSCD Configuration on Cosmo v9 Public Security Target**
FQR No	**110 9320**
FQR Issue	**3**

DOCUMENT REVISION

Date	Revision	Modification
14/10/2019	1.0	Creation based on the full ST
29/10/2019	2.0	Update AGD version
20/11/2019	3.0	Review and Update

TABLE OF CONTENTS

1 Definitions

ADF	Application Dedicated File
AES	Advanced Encryption Standard
AID	Application Identifier
APDU	Application Protocol Data Unit (command received/Data sent by the chip)
API	Application Programming Interfaces
CA	Certification authority
CBC	Cipher Block Chaining
CGA	Certificate Generation Authority (Authority in charge of generating the qualified certificate(s))
C/S	Client / Server
DAP	Data Authentication Pattern
CSP	Certificate Service Provider
DES	Data Encryption Standard
DF	Dedicated File
DH	Diffie Hellman
DTBS	Data to be signed (Sent by the SCA)
DTBS/R Representation	Representation of the Data to be signed
EAL	Evaluation Assurance Level
EF	Elementary File
GP	Global Platform
HID	Human Interface Device
IC	Integrated Chip
MAC	Message Authentication code
PIN	Personal Identification Number
PKI	Public Key Infrastructure
RAD	Reference Authentication Data (PIN stored called also PIN_{Sig})
RSA	Rivest Shamir Adleman
SCA	Signature creation Application
SCD	Signature Creation Data
SCP	Secure Channel Procotol
SHA	Secure hashing Algorithm
SSCD	Secure Signature Creation Device
Sub-CA	Subordinate Certificate Authority
SVD	Signature Verification Data
TOE	Target of evaluation
URL	Uniform Resource Locator
USB	Universal Serial Bus
VAD	Verification Authentication Data (PIN submitted by the holder)
XML	eXtensible Markup Language

2 References

[AGD_PRE] FQR 220 1306 – CombICAO Applet – Perso Guide, Ed 8. IDEMIA

[AGD_OPE] FQR 220 1307 – CombICAO Applet – User Guide, Ed 9. IDEMIA

[ANSIX9.31] "Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (DSA)" - ANSI X9.31-1998, American Bankers Association

[ANSIX9.62] ANSI x9.62-2005 Public Key Cryptography for the Financial Services Industry – The Elliptic Curve Digital Signature Algorithm (ECDSA)

[AN10] JIL - Certification of "open" smart card products - Version 1.1 - 4 February 2013

[CC31-1] "Common Criteria for information Technology Security Evaluation, Part 1: Introduction and general model", April 2017, Version 3.1 revision 5

[CC31-2] "Common Criteria for information Technology Security Evaluation, Part 2: Security Functional requirements", April 2017, Version 3.1 revision 5

[CC31-3] "Common Criteria for information Technology Security Evaluation, Part 3: Security Assurance requirements", April 2017, Version 3.1 revision 5

[Directive] Directive 1999/93/EC of the european parliament and of the council of 13 December 1999 on a community framework for electronic signatures

[eIDAS] REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

[GP2.3] Global Platform, Card Specification - Version 2.3 – October 2015.

[IEEE] IEEE Std 1363a-2004 Standard Specification of Public-Key Cryptography

[ISO_15946] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves

[JIL-COMP] Joint Interpretation Library - Composite product evaluation for Smart Cards and similar devices – v1.2

[PKCS#1] PKCS #1 v2.1: RSA Cryptography Standard - June 14, 2002

[PKCS#3] PKCS#3 - Diffie-Hellman Key-Agreement Standard - Version 1.4, November 1, 1993*

[PLT] ID-One COSMO V9 Essential Platform certified by NSCIB under reference CC-18-200833

[PP_IC] Security IC Platform Protection Profile with augmentation packages - Version 1.0 - BSI-CC-PP-0084-2014

[PTF_AGD1] ID-One Cosmo V9 Application Loading Protection Guidance, FQR 110 8798, Issue 2. IDEMIA

[PTF_AGD2] ID-One Cosmo V9 Applet Security Recommendations, FQR 110 8794, Issue 4. IDEMIA

[PTF_AGD_OPE] ID One Cosmo V9.0 Essential Reference Guide, 22 October 2018, FQR 110 8823, Ed5. IDEMIA

[PTF_AGD_PRE] ID One Cosmo V9.0 Essential - Pre-Perso Guide, FQR 110 8797 Ed5 AGD PRE. IDEMIA

[PTF_AGD_SEC_AC] Secure acceptance and delivery of sensitive element - FQR 110 8921 Ed1, IDEMIA

[SCP03] Global Platform Card Technology, Secure Channel Protocol 03, Card Specification v 2.2 - Amendment D - Version 1.1 - September 2009.

[SSCD2] Protection profiles for secure signature creation device — Part 2: Device with key generation Version 2.0.1 – 23/01/2012 – Reference BSI-CC-PP-0059-2009-MA-01

[SSCD3] Protection profiles for secure signature creation device — Part 3: Device with key import Version 1.0.2 – 24/07/2012 – Reference BSI-CC-PP-0075

[SSCD4] Protection profiles for secure signature creation device — Part 4: Extension for device with key generation and trusted communication with certificate generation application Version 1.0.1 – 14/11/12 – Reference BSI-CC-PP-0071

[TR_03110] TR 03110 v2.10 Advanced Security Mechanisms for Machine Readable Travel Documents, Part 1, Part 2 and Part 3. (2012)

[TR_03111] Technical Guideline TR-03111 - Elliptic Curve Cryptography - Version 2.0

[CEN_14890] CEN/EN 14890:2013 Application Interface for smart cards used as Secure Signature Creation

[7816-4] ISO/IEC 7816-4:2013, Identification Cards — Integrated circuit cards— Part 4 : Organization, security and commands for interchange

3 Security Target Introduction

3.1 Public Security Target Reference

Title	CombICAO Applet in SSCD configuration on Cosmo v9 – Public Security Target
Reference and version	FQR 110 9320, version 3
Author	IDEMIA
Certification Body	NSCIB
CC version	3.1 revision 5
EAL	EAL5 augmented with AVA_VAN.5 and ALC_DVS.2
Protection Profiles	PP SSCD-Part 2 Key Generation [SSCD2], PP SSCD-Part 3 Key Import [SSCD3], PP SSCD-Part 4 Key Generation and Trusted Channel with CGA [SSCD4]

3.2 TOE Reference

Product name	CombICAO Applet
TOE name	CombICAO Applet in SSCD configuration on Cosmo v9
Developer name	IDEMIA
TOE identification	SAAAAR 203297
Platform reference	ID-ONE Cosmo V9 Essential version 3 (Cosmo V9) (certified by the Dutch NSCIB certification body (CC-18-200833) on 14-12-2018)
Platform Identification	089233
IC reference	Infineon smart card IC (Security Controller) IFX_CCI_000003h, 000005h, 000008h, 00000Ch, 000013h, 000014h, 000015h, 00001Ch, 00001Dh, 000021h, 00022Dh, design step H13 with optional libraries CCL V2.0.0002, RSA2048/4096 V2.07.003 / V2.06.003, EC V2.07.003 / V2.06.003, Toolbox V2.07.003 / V2.06.003, HSL V02.01.6634 / V01.22.4346, MCS V02.02.3389 / V02.03.3446, SCL V2.02.010 and with specific IC dedicated software (certified by the German BSI certification body (BSI-DSZ-CC-0945-V2-2018) on 20-04-2018)
Guidance documents	[AGD_PRE] and [AGD_OPE] [PTF_AGD_OPE], [PTF_AGD1], [PTF_AGD2], [PTF_AGD_SEC_AC] and [PTF_AGD_PRE]

The TOE identification is described in [AGD_PRE].

3.3 TOE overview

3.3.1 TOE Type

The CombICAO Applet is a configurable applet designed primarily for identification, authentication, signature and seal generation, and as a machine readable travel document (MRTD). This public security target focuses only on Qualified Signature or Seal Creation Device (QSCD) configuration, used to create advanced or qualified signature or seal in the sense of [eIDAS]. MRTD configurations are managed in others security targets.

The TOE is a composite product made up of an embedded software developed using javacard technology, composed on a javacard open platform. Both are developed by IDEMIA.

The javacard open platform has already been certified. For more details see [PLT].

The embedded software is made up the javacard CombICAO applet [Applet], which relies on Javacard API provided by the underlying javacard open platform.

3.3.2 TOE scope

The TOE is made up of:

- The underlying javacard open platform
- The javacard CombICAO code [Applet]
- The associated guidance documentation in [AGD_PRE] and [AGD_OPE].

Moreover, as the [PLT] is certified as a javacard open platform and complies with the requirements of the Application note 10 [AN10], and as the TOE complies also with [AN10], the TOE may also contain any other applets that comply with [AN10] and the specific requirements of the TOE stated in the guidance documents.

The TOE scope is shown in figure 1. Once constructed, the TOE is a bare microchip with its external interfaces for communication. The physical medium on which the microchip is mounted does not alter nor modify any security functions of the TOE.

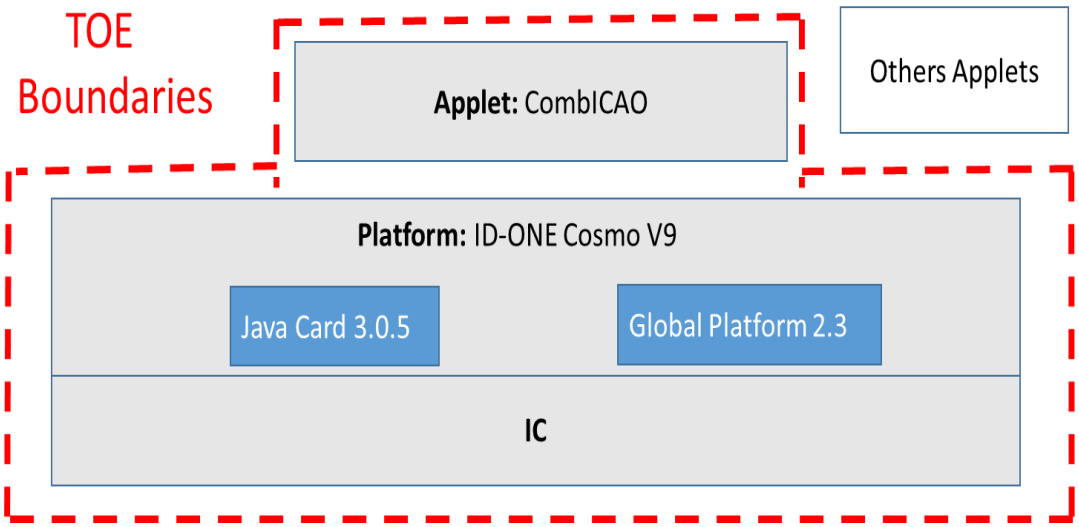


Figure 1 - TOE scope

3.3.3 Required non-TOE hardware/software/firmware

The TOE is a Qualified Signature Creation Device. It is an independent product and does not need any additional hardware/software/firmware to ensure its security.

In order to be powered up and to be able to communicate the TOE needs a reader.

3.3.4 Usage and major security features

The TOE intended usage is to be used as a “qualified signature creation device” with key generation and/or key import, with respect to the [eIDAS].

Within the framework described by [SSCD2], [SSCD3], and [SSCD4], the TOE allows to

- perform basic, advanced and qualified signature;
- authenticate the signatory thanks to PIN verification;

- authenticate one (or several) administrator(s) of the TOE, that may have special rights to administrate the SCD and SVD (generation, import);
- establish trusted channel, protected in integrity, authenticity and confidentiality, with trusted IT entities such as a CGA;
- Secure execution of services.

The scope of [SSCD2], [SSCD3], and [SSCD4] is extended in several ways:

- Personalization phase including:
  - authentication protocol;
  - access control;
  - encryption mechanism involved in key loading;
  - initialization of the data structure;
  - data loading;
  - phase switching.
- All authentication protocols (PACE-GM/IM/CAM) and secure messaging type (DES-128,AES128/192/256);
- All supported digital signature algorithm;
- Authentication of the TOE using asymmetric cryptography;
- All PIN management operations available after delivery point (spanning the three types of PIN : PIN_{Auth}, PIN_{Sig} (called also RAD) & PUK):
  - PIN initialization;
  - Upgrade of PIN attributes;
  - PIN change, unlocking, (re-)initialization;
  - Certificate management.
- PACE authentication;
- Extended Access Control Version 1 as defined in [TR-03110]. It consists of two parts: Chip Authentication Protocol Version 1 and Terminal Authentication Protocol Version 1;
- Signature key import in personalization phase;
- Signature key generation in personalization phase and use phase;
- Signature key public key export in personalization phase and use phase;
- Digital authentication feature including (1) the corresponding key management operation (generation, import), and (2) security policies applicable to each of these operations (authentication, generation, import);

The TOE may be used for various use cases requiring qualified signature:

- Digital signature application;
- Electronic health card;
- Electronic services cards;
- ……;

Depending on the use case and or the ability of the underlying javacard open platform, the TOE may be used

- in contact mode (T=0 and/or T=1 protocol);
- in contactless protocol (T=CL);

3.4 TOE Description

The TOE is compliant with the specification [CEN_14890], with the following types of data structures:

- files, compliant with [7816-4];
- keys;
- PINs;

The TOE handles the following types of file (described in [7816-4]):

- Transparent File – also named Elementary File (EF);
- Application Dedicated File (ADF);
- Master File (MF);

All these files are organized within a File System compliant to [7816-4]. It represents the hierarchy between all the files. At the top of the structure stands the Master File, it is the default selected file at reset. Under the Master File, are located the Application Dedicated File(s). The Master File, as well as each ADF, may contain Elementary File, keys and/or PINs.

Each file is characterized by its own attributes, such as:

- Access conditions for read and write access (for EF) or selection (for ADF);
- File identifier;
- Location within the File System;
- Size (for EF);

The TOE allows to:

- create two types of file (Application Dedicated File and Elementary File), which updates the File System;
- read, update, resize any Elementary File;
- move within the File Structure by use of file selection;

3.4.1 Keys and PINs

The TOE handles as well cryptographic data objects, such as keys (for digital signature, authentication, and encryption key decipherment) and PINs.
The TOE enables to create, update and use PINs as detailed in [AGD_OPE].
For keys, the TOE enables to create, import, generate and erase keys as detailed in [AGD_OPE].

3.4.2 Access Control Management

The TOE ensures access control on any operations acting on any objects it handles (files, keys or PINs).

Each EF is configured at creation with access conditions protecting read and write access, while the ADF may be configured at creation with access conditions protecting their selection. Keys used for digital authentication, digital signature creation, encryption key decipherment and PINs require specific conditions before they can be used, updated or managed.

Prior to granting access to a given operation, the TOE checks the requested access rights are fulfilled. The access conditions can only be fulfilled upon successful authentication of an entity (see below).

3.4.3 Authentication of entities

The TOE allows authenticating several types of entities in order to grant them some access rights:

- **Authentication of a natural person**. It relies on a successful verification of a PIN code presented to the TOE by the natural person. (only available in phase 7)
- **Authentication of a remote server**. It relies on a mutual authentication - based on PKI - generating a trusted channel ensuring authenticity, integrity and confidentiality of the messages, used to securely communicate. (only available in phase 7)
- **Authentication of personalization agent** (only in phase 6);

These authentication mechanisms allows fulfilling the access control mechanisms described above.

3.4.4 Digital authentication

The TOE supports digital authentication based on RSA and elliptic curves cryptography (ECC). Digital authentication is the process by which (1) the holder of TOE authenticates itself to the TOE using a PIN, releasing access right to an authentication key stored in the TOE, (2) subsequently the authentication key is used by the TOE to authenticate itself on behalf of the TOE holder. Digital authentication is useful so that the TOE holder can authenticate himself on line, without compromising any sensitive assets (PINs or authentication key).

3.4.5 Electronic Services

The TOE supports as well several electronic services:

- **Digital signature (or seal)**: this feature enables the signatory to electronically signs (or seals) documents. The signature (or seal) may be either advanced or qualified (compliant with [SSCD2] and [SSCD3]).
- **Encryption key decipherment**: this feature enables the document holder to store secret data on an electronic vault. The key needed to decipher the key encrypting these data is securely stored in the TOE. The document holder's computer sends the encrypted encryption key to the TOE to get the plain encryption key.

3.4.6 Secure execution

The TOE ensures a secure execution of its services. First, the TOE ensures its execution is protected against physical manipulation or attempt to tamper with. Secondly, should the execution of the TOE be tampered with in any manner, the TOE ensures it remains in a safe state protecting its assets and the TSFs, so that no vulnerabilities can be exploited by an attacker.

3.5 Life Cycle

3.5.1 Life cycle overview

The TOE life cycle in the figure 2 distinguishes stages for development, production, preparation and operational use in accordance with the standard smart card life cycle [PP_IC].

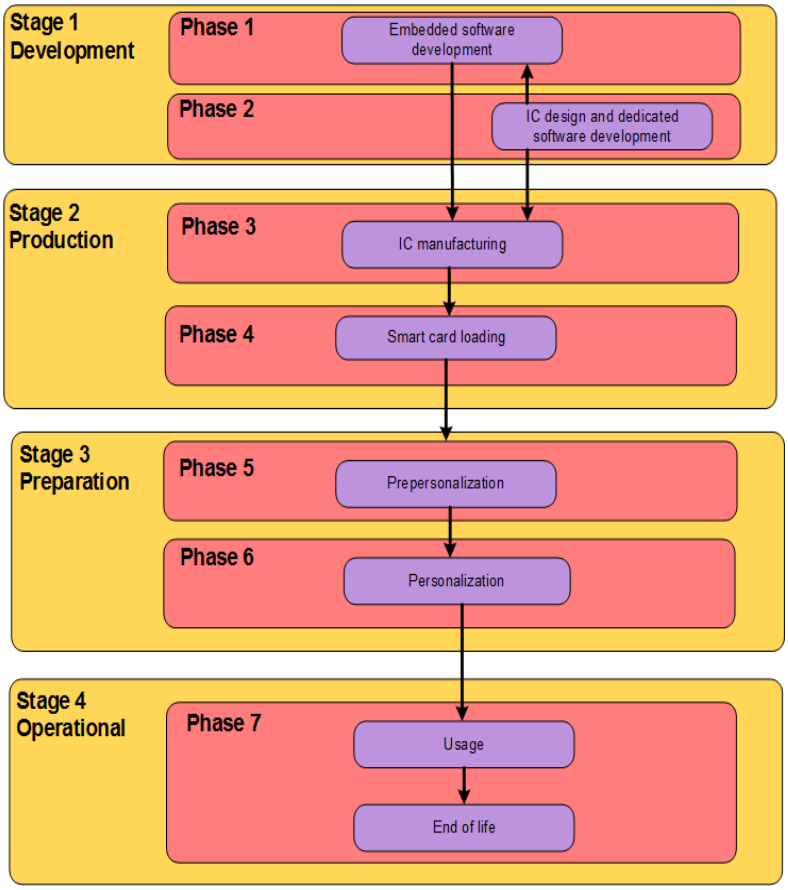


Figure 2 - Life Cycle Overview

3.5.2 Development Environment

In this environment, the following two phases take place:

- Phase 1: IC Embedded Software Development (Javacard Open Platform components and CombICAO applet)
- Phase 2: IC Development

The IC Embedded Software Developer is in charge of the specification, development and validation of the software (Javacard Open Platform and CombICAO applet).

The IC Developer designs the IC, develops the IC dedicated software and provides information, software or tools to the IC embedded software developer.

Roles, actors, sites and coverage for this environment of the product life-cycle are listed in the table below:

Role	Actor	Site	Covered by
CombICAO Applet Developer	IDEMIA	MANILA and Courbevoie R&D sites	ALC
Platform Developer	IDEMIA	IDEMIA R&D sites Refer to [PLT]	ALC
IC Developer	Infineon	Infineon R&D sites Refer to [PLT]	ALC

Table 1 Roles, actors, sites and coverage for the 3.5.2 development environment

3.5.3 Production Environment

In this environment, the following two phases take place:

- Phase 3: IC manufacturing
- Phase 4: Smart card loading

The IC manufacturer is responsible for producing the IC (manufacturing, testing, initialization). Depending on the intention:

- **(Option 1)** the developer sends the image (containing both the javacard platform and the CombICAO applet) to be flashed in the IC to the IC manufacturer in the phase 3.

Or

- **(Option 2)** the platform developer sends the image (containing only the javacard platform) to be flashed in the IC to the IC manufacturer in the phase 3. Once the javacard platform has been loaded, the package of CombICAO is securely delivered from the applet developer to the smart card loader. The cap file of the applet is then loaded (using GP) in the javacard platform by the smart card loader in phase 4 at IDEMIA audited site.

Or

- **(Option 3)** the developer sends the image (containing both the javacard platform and the CombICAO applet) to be loaded in Flash (using the loader of the IC) to the smart card loader in phase 4.

Several life cycles are available, depending when the Flash Code is loaded. The following tables present roles, actors, sites and coverage for this for this environment of the product life-cycleand describe for each of them the TOE delivery point.

Role	Package to be loaded	Actor	Site	Covered by
IC manufacturer	Image containing both javacard platform and applet	IC manufacturer	IC manufacturer production plants Refer to [PLT]	ALC
Smart card loader	-	-	-	-
TOE Delivery Point				

Table 2 Image contained both platform and applet is loaded at IC manufacturer (Option 1)

Role	Package to be loaded	Actor	Site	Covered by
IC manufacturer	Image containing only javacard platform	IC manufacturer	IC manufacturer production plants Refer to [PLT]	ALC
Smart card loader	Cap file of the applet	IDEMIA	IDEMIA plants (Shenzhen, Haarlem, Vitré)	ALC
TOE Delivery Point				

Table 3 Cap file of CombICAO applet is loaded through the loader of the IC manufacturer (Option 2)

Role	Package to be loaded	Actor	Site	Covered by
IC manufacturer	-	-	-	-
TOE Delivery Point				
Smart card loader	Image containing both javacard platform and applet	IDEMIA or another agent	IDEMIA plants or others sites	AGD

Table 4 Image contained both platform and applet is loaded through the loader of the IC (Option 3)

The following table describes the physical delivery of the TOE components from ALC phase to AGD phase:

TOE component	Identification	Package	Delivery method
CombICAO applet in SSCD configuration on Cosmo v9	SAAAAR 203297	The package can be either of the following: - Image contained both platform and applet, - Chip embedded in ID1 cards or ID3 holder pages, - Chip embedded in antenna inlays, - Chip in modules.	CPS tool is used in the case of an Image delivery. Otherwise, trusted courier is used.
[AGD_PRE]	FQR 220 1306, Ed 8	Electronic document	PGP-encrypted email
[AGD_OPE]	FQR 220 1307, Ed 9	Electronic document	PGP-encrypted email
[PTF_AGD1]	FQR 110 8798, Issue 2	Electronic document	PGP-encrypted email
[PTF_AGD2]	FQR 110 8794, Issue 4	Electronic document	PGP-encrypted email
[PTF_AGD_OPE]	FQR 110 8823, Ed5	Electronic document	PGP-encrypted email
[PTF_AGD_PRE]	FQR 110 8797, Ed5	Electronic document	PGP-encrypted email
[PTF_AGD_SEC_AC]	FQR 110 8921, Ed1	Electronic document	PGP-encrypted email

Table 5 Physical delivery of the TOE components from ALC phase to AGD phase

3.5.4 Preparation Environment

In this environment, the following two phases take place:

- Phase 5: Pre-personalization
- Phase 6: Personalization

The preparation environment may not necessarily take place in a manufacturing site, but may be performed anywhere. All along these two phases, the TOE is self-protected as it requires the authentication of the pre-personalization agent or personalization agent prior to any operation. The CombICAO applet is pre-personalized and personalized according to [AGD_PRE].

At the end of phase 6, the TOE is constructed. These two phases are covered by [AGD_PRE] tasks of the TOE and AGD_OPE tasks of [PLT]. Notice that all security features related to the pre-personalization phase are covered by the underlying platform [PLT].

3.5.5 Operational Environment

The TOE is under the control of the User (Signatory and/or Administrator).

During this phase, the TOE may be used as described in §3.4. This phase is covered by [AGD_OPE] tasks of the TOE and AGD_OPE tasks of [PLT].

4 Conformance Claim

4.1 CC and package Conformance claim

This public security target claims conformance to the Common Criteria version 3.1, revision 5 ([CC31-1], [CC31-2] and [CC31-3]).

The conformance to the Common Criteria is claimed as follows:

CC	Conformance rationale
Part 1	Strict Conformance
Part 2	Conformance extended with ▪ FCS.RND.1: "Quality Metric for Random Numbers" ▪ FPT_EMS.1: "TOE Emanation" ▪ FIA_API.1: "Authentication proof of Identity" ▪ FMT_LIM.1 Limited capabilities ▪ FMT_LIM.2 Limited availability
Part 3	Conformance to assurance package EAL 5, augmented with ▪ AVA_VAN.5: "Advanced methodical vulnerability analysis" ▪ ALC_DVS.2: "Sufficiency of security measures"

Moreover the security target claims compliance with application note 10 [AN10].

4.2 PP Conformance Claim

This security target claims a **strict** conformance to the Secure Signature Creation Device (SSCD) Protection Profile [SSCD2], [SSCD3] conform to CC version 3.1 revision 3 and [SSCD4] conform to CC version 3.1 revision 4.

This security target also addresses the manufacturing and personalization phases at TOE level (cf. TOE life cycle presented in §3.5. These additions do not contradict any of the threats, assumptions, organizational policies, objectives or SFRs stated in the protection profiles that cover the operational phase of the signature device.

Additional information are stated in the following chapter.

4.3 Conformance rationale

4.3.1 Additional assets

All assets from the protection profiles are included in this security target. Other assets have been added (see section 5.1.1.2).

4.3.2 Additional Roles

The roles from protection profiles are maintained in this security target. Other roles have been added (see section 5.1.2.2).

4.3.3 Additional threats

All the threats from the protection profiles are maintained in this security target. Other threats have been added (see section 5.2.2).

4.3.4 Additional OSPs

All the Policies from the protection profiles are maintained in this security target. Other OSPs have been added (see section 5.3.2).

4.3.5 Additional objectives

4.3.5.1 Additional Security objectives for the TOE

All the security objectives for the TOE from the protection profiles are maintained in this security target. Other security objectives for the TOE have been added (see section 6.1.2).

4.3.5.2 Additional Security objectives for the Operational Environment

All the security objectives for the operational environment from the protection profiles are maintained in this security target. Other security objectives for the operational environment have been added (see section 6.2.2).

4.3.6 Additional SFRs

All the SFRs from the protection profiles are maintained. Other SFRs have been added (see section 8.1.3).to cover supplemental features.

4.3.7 Package conformance

The protection profiles require an assurance level of level EAL4 augmented with AVA_VAN.5.
This security target considers an assurance level EAL5 augmented with AVA_VAN.5 and ALC_DVS.2, which still complies with the requirements of the protection profiles.

5 Security Problem Definition

5.1 Assets and users

5.1.1 Assets

5.1.1.1 Assets from protection profiles

- **SCD**: private key used to perform an electronic signature operation. The confidentiality, integrity and signatory's sole control over the use of the SCD must be maintained.
- **SVD**: public key linked to the SCD and used to perform electronic signature verification. The integrity of the SVD when it is exported must be maintained.
- **DTBS** and **DTBS/R**: set of data, or its representation, which the signatory intends to sign. Their integrity and the unforgeability of the link to the signatory provided by the electronic signature must be maintained.

5.1.1.2 Additional Assets

- **Keys**:
  - Private or secret key(s) used to (1) authenticate an external user or entity, (2) perform authentication protocols, (3) perform digital authentication, (4) perform digital signature or (5) perform encryption key decipherment. Their integrity and confidentiality must be maintained.
  - Public key(s) used as trust anchor to verify a certificate chain used in terminal authentication. Their integrity must be maintained.
- **PIN/PUK:** The applet shall manage two types of PINs (PIN_{Sig} called also RAD and PIN_{Auth}) for user authentication and one PUK for management purpose. They are used to authenticate natural persons. The PINs and PUK must be created and initialized first before they can be used for authentication.
- **VAD:** PIN code entered by the end user to perform a signature operation (confidentiality and authenticity of the VAD as needed by the authentication method employed)
- **Session keys:** Keys computed for secure messaging and used to ensure confidentiality, authenticity and integrity of data.
- **Authenticity of the Electronic Documents Chip**: The authenticity of the electronic document's chip, personalized by the issuing organization for the Document holder, is used by the electronic document presenter to prove his possession of a genuine electronic document.
- **Tracing Data**: Technical information about the current and previous locations of the electronic document gathered unnoticeable by the Document holder recognizing the TOE not knowing any PACE password. TOE tracing data can be provided / gathered.
- **Sensitive User Data**: User data, which have been classified as sensitive data by the electronic document issuer. Sensitive user data are a subset of all user data, and are protected by EAC.
- **User Data stored on the TOE**: All data, with the exception of authentication data, that are stored in the context of the application(s) on the electronic document. These data are allowed to be accessed either by a BAT, or, in the case of sensitive data, by an Authentication Terminal with appropriate authorization level.

- **User Data transferred between the TOE and the Terminal**: All data, with the exception of authentication data, that are transferred (both directions) during usage of the application(s) of the electronic document between the TOE and authenticated terminals.

- **Accessibility of TOE Functions and Data only for Authorized Subjects**: Property of the TOE to restrict access to TSF and TSF-Data stored in the TOE to authorized subjects only.

- **Genuineness of the TOE**: Property of the TOE to be authentic in order to provide claimed security functionality in a proper way.

- **Electronic Document Communication Establishment Authorization Data**: Restricted-revealable authorization information for a human user used for verification of the authorization attempts as an authorized user (PACE password). These data are stored in the TOE and not send to it. Restricted-revealable here refers to the fact that if necessary, the Document holder may reveal her verification values of CAN to an authorized person, or to a device that acts according to respective regulations and is considered trustworthy.

- **Secret Document holder Authentication Data:** Secret authentication information for the Document holder being used for verification of the authentication attempts as authorized Document holder (sent PACE passwords, e.g. PIN, PUK or CAN).

- **TOE internal Non-Secret Cryptographic Material:** Permanently or temporarily stored non-secret cryptographic (public) keys and other non-secret material used by the TOE in order to enforce its security functionality.

5.1.2 Subjects

5.1.2.1 Subjects from protection profiles

- **User**: End user of the TOE who can be identified as administrator or signatory. The subject S.User may act as S.Admin in the role R.Admin or as S.Sigy in the role R.Sigy.

- **Administrator**: User who is in charge to perform the TOE initialisation, TOE (pre-) personalization or other TOE administrative functions. The subject S.Admin is acting in the role R.Admin for this user after successful authentication as administrator.

  Note
  For all activities related to Personalization, the subject Administrator is called also Personalization Agent in the rest of the document.

- **Signatory**: User who holds the TOE and uses it on its own behalf or on behalf of the natural or legal person or entity they represent. The subject S.Sigy is acting in the role R.Sigy for this user after successful authentication as signatory.

- **Attacker**: Human or process acting on their behalf located outside the TOE. The main goal of the attacker is to access the SCD or to falsify the electronic signature. The attacker has got a high attack potential and knows no secret.

5.1.2.2 Additional subjects:

- **Issuer Certification Authority (Issuer CA)**: An organization enforcing the policy of the electronic document issuer, i. e. confirming correctness of user and TSF data that are stored within the electronic document. The Issuer

CA represents both the (1) root and (2) intermediate sub-CAs of the public key infrastructure (PKI) used to issue the electronic document. The Issuer CA signs user and TSF data to create a digital seals that is stored in the electronic document to demonstrate their integrity and authenticity.

- **Country Verifying Certification Authority (CVCA**): The Country Verifying Certification Authority (CVCA) enforces the privacy policy of the issuing state or organization, i. e. enforcing protection of sensitive user data that are stored in the electronic document. The CVCA represents the country specific root of the PKI of Authentication Terminals and eServices certification authorities, and creates eServices certification authorities certificates within this PKI. Updates of the public key of the CVCA are distributed as CVCA Link-Certificates, see [TR-03110].
- **eService certification authority**: An organization issuing terminal certificates. The eService certification authority is a certificate authority, authorized by the corresponding CVCA to issue certificates for Authentication Terminals.
- **Document holder:** A person who the electronic document issuer has personalized the electronic document for. Personalization here refers to associating a person uniquely with a specific electronic document. Note that an Document holder can also be an attacker. The document holder is equivalent to the signatory and can use and manage the PIN_{Sig} (called also the RAD), the PIN_{Auth} and the PUK.
- **Electronic document presenter**: A person presenting the electronic document to a terminal and claiming the identity of the Document holder. Note that an electronic document presenter can also be an attacker.
- **Basic Authentication Terminal (BAT):** A BAT implements the terminal part of the PACE protocol and/or the VERIFY PIN command and authenticates itself to the electronic document using a shared password (CAN, PIN, PUK). A BAT is not allowed to access sensitive user data.
- **Authentication Terminal**: A terminal that has successfully passed Terminal Authentication is an Authentication Terminal. It is authorized by the electronic document issuer through the eServices certification authorities of the receiving branch (by issuing terminal certificates) to access a subset or all of the data stored on the electronic document.
- **Terminal**: A terminal is any technical system communicating with the TOE through the contactless or contact-based interface. The role terminal is the default role for any terminal being recognized by the TOE that is neither a BAT nor an Authentication Terminal.

5.2 Threats

5.2.1 Threats drawn from the protection profiles

5.2.1.1 T.SCD_Divulg *Storing, copying and releasing of the signature creation data*

An attacker stores or copies the SCD outside the TOE. An attacker can obtain the SCD during generation, storage and use for signature creation in the TOE.

5.2.1.2 T.SCD_Derive *Derive the signature creation data*

An attacker derives the SCD from publicly known data, such as SVD corresponding to the SCD or signatures created by means of the SCD or any other data exported outside the TOE, which is a threat against the secrecy of the SCD.

5.2.1.3 T.Hack_Phys *Physical attacks through the TOE interfaces*

An attacker interacts physically with the TOE to exploit vulnerabilities, resulting in arbitrary security compromises. This threat is directed against SCD, SVD and DTBS.

5.2.1.4 T.SVD_Forgery *Forgery of the signature verification data*

An attacker forges the SVD presented by the CSP to the CA. This results in loss of SVD integrity in the certificate of the signatory.

5.2.1.5 T.SigF_Misuse *Misuse of the signature creation function of the TOE*

An attacker misuses the signature creation function of the TOE to create SOD for data the signatory has not decided to sign. The TOE is subject to deliberate attacks by experts possessing a high attack potential with advanced knowledge of security principles and concepts employed by the TOE.

5.2.1.6 T.DTBS_Forgery *Forgery of the DTBS/R*

An attacker modifies the DTBS/R sent by the SCA. Thus the DTBS/R used by the TOE for signing does not match the DTBS the signatory intended to sign.

5.2.1.7 T.Sig_Forgery *Forgery of the electronic signature*

An attacker forges a signed data object, maybe using an electronic signature, which has been created by the TOE, and the violation of the integrity of the signed data object is not detectable by the signatory or by third parties. The signature created by the TOE is subject to deliberate attacks by experts possessing a high attack potential with advanced knowledge of security principles and concepts employed by the TOE.

5.2.2 Additional threats

5.2.2.1 T.Key_Divulg *Storing, copying, and releasing of a key stored in the TOE*

An attacker can store and copy a key (other than SCD) stored in the TOE outside the TOE. An attacker can release a key during generation, storage and use in the TOE.

5.2.2.2 T.Key_Derive Derive a key

An attacker derives a key (other than SCD) from public known data, such as the corresponding public key or cryptogram created by means of the key or any other data communicated outside the TOE, which is a threat against the secrecy of the key.

5.2.2.3 T.TOE_PublicAuthKey_Forgery *Forgery of the public key of a TOE authentication key*

An attacker forges the public key of a TOE authentication key presented by the TOE. This results in loss of the public key integrity in the authentication certificate of the TOE.

5.2.2.4 T.Authentication_Replay *Replay of an authentication of an external entity*

An attacker retrieves by observation authentication data used by a third party during an authentication sequence. The attacker tries to replay this authentication sequence to grant access to the TOE.

5.2.2.5 T.Counterfeit

An attacker with high attack potential produces an unauthorized copy or reproduction of a chip of a genuine electronic document. This copy or reproduction can be used as a part of a counterfeit electronic document. This violates the authenticity of the electronic document's chip used for authentication of an electronic document presenter by possession of an electronic document. The attacker may generate a new data set or extract completely or partially the data from a genuine electronic document’s chip and copy them to another appropriate chip to imitate the chip of the genuine electronic document.
Threat agent: having high attack potential, being in possession of one or more legitimate ID-Cards

5.2.2.6 T.Sensitive_Data

An attacker tries to gain access to sensitive user data through the communication interface (contact or contactless) of the electronic document’s chip. The attack T.Sensitive_Data is similar to the threat T.Skimming from [PACEPP] w.r.t. the attack path (communication interface) and the motivation (to get data stored on the electronic document’s chip) but differs from those in the asset under the attack, the opportunity (i.e. knowing the PACE Password or the PIN) and therefore the possible attack methods.
Threat agent: having high attack potential, knowing the PACE Password or the PIN, being in possession of a legitimate electronic document

5.2.2.7 T.Abuse-Func

An attacker may use functions of the TOE which shall not be used in TOE operational phase in order (i) to manipulate or to disclose the User Data stored in the TOE, (ii) to manipulate or to disclose the TSF-data stored in the TOE or (iii) to manipulate (bypass, deactivate or modify) soft-coded security functionality of the TOE. This threat addresses the misuse of the functions for the initialization and Personalization in the operational phase after delivery to the holder.

5.2.2.8 T.Eavesdropping

An attacker is listening to the contactless communication between the electronic document and the BAT in order to gain the user data transferred between the TOE and the terminal connected.

5.2.2.9 T.Forgery

An attacker fraudulently alters the User Data or/and TSF-data stored on the electronic document or/and exchanged between the TOE and the terminal connected in order to outsmart the BAT by means of changed electronic document holder's related reference data. The attacker does it in such a way that the terminal connected perceives these modified data as authentic one.

5.2.2.10 T.Information_Leakage

An attacker may exploit information leaking from the TOE during its usage in order to disclose confidential User Data or/and TSF-data stored on the electronic document or/and exchanged between the TOE and the terminal connected. The information leakage may be inherent in the normal operation or caused by the attacker.

5.2.2.11 T.Malfunction

An attacker may cause a malfunction the electronic document's hardware and Embedded Software by applying environmental stress in order to (i) deactivate or modify security features or functionality of the TOE' hardware or to (ii) circumvent, deactivate or modify security functions of the TOE's Embedded Software. This may be achieved e.g. by operating the electronic document outside the normal operating conditions, exploiting errors in the electronic document's Embedded Software or misusing administrative functions. To exploit these vulnerabilities an attacker needs information about the functional operation.

5.2.2.12 T.Phys-Tamper

An attacker may perform physical probing of the electronic document in order (i) to disclose the TSF-data, or (ii) to disclose/reconstruct the TOE's Embedded Software. An attacker may physically modify the electronic document in order to alter (i) its security functionality (hardware and software part, as well), (ii) the User Data or the TSF-data stored on the travel document.

5.2.2.13 T.Skimming

An attacker imitates a terminal in order to get access to the user data stored on or transferred between the TOE and the inspecting authority connected via the contactless interface of the TOE.

5.2.2.14 T.Tracing

An attacker tries to gather TOE tracing data (i.e. to trace the movement of the electronic document) unambiguously identifying it remotely by establishing or listening to a communication via the contactless interface of the TOE.

5.3 Organisational Security Policies

5.3.1 Security policies drawn from the protection profiles

5.3.1.1 P.CSP_QCert *Qualified certificate*

The CSP uses a trustworthy CGA to generate a qualified certificate or non-qualified certificate (cf. [directive], article 2, clause 9, and Annex I) for the SVD. The certificates contain at least the name of the signatory and the SVD matching the SCD implemented in the TOE under sole control of the signatory. The CSP ensures that the use of the TOE as SSCD is evident with signatures through the certificate or other publicly available information.

5.3.1.2 P.Qsign *Qualified electronic signatures*

The signatory uses a signature creation system to sign data with an advanced electronic signature (cf. the directive, article 1, clause 2), which is a qualified electonic signature if it is based on a valid qualified certificate (according to the directive, annexe I)[1]. The DTBS are presented to the signatory and sent by the SCA as DTBS/R to the SSCD. The SSCD creates the electonic signature created with a SCD implemented in the SSCD that the signatory maintain under their sole control and is linked to the DTBS/R in such manner that any subsequent change of the data is detectable.

5.3.1.3 P. Sigy_SSCD *TOE as secure signature creation device*

The TOE meets the requirements for an SSCD laid down in Annex III of the directive. This implies the SCD is used for digital signature creation under the sole control of the signatory and the SCD can pratically occur only once.

5.3.1.4 P.Sig_Non-Repud *Non-repudiation of signatures*

The lifecycle of the SSCD, the SCD and the SVD shall be implemented in a way that the signatory is not able to deny having signed data if the signature is successfully verified with the SVD contained in their unrevoked certificate.

5.3.2 Additional security policies

5.3.2.1 P.LinkSCD_QualifiedCertificate *Link between a SCD stored in the TOE and the relevant qualified certificate*

The Role in charge of creating and updating the SCD (**Personalization Agent**, **R.Admin**), or the trusted IT entity involved in the updating process (CSP**)** shall ensure an unambiguous link between the (qualified) certificate(s) and the corresponding SCD(s). This link might be figured out by a PKCS#15 structure, an XML structure, an identifier .linking the file containing the (qualified) certificate or the URL hosting them to the SCD(s) stored in the TOE. In particular, it implies this link is updated, each time the SCD(s) is created, imported, erased or generated.

5.3.2.2 P.TOE_PublicAuthKey_Cert *Certificate for asymmetric TOE authentication keys*

The TOE contains certificate(s) issued by a known entity ensuring its public key corresponding to the authentication private key is genuine.

5.3.2.3 P.eServices *Provision of eServices*

The TOE provides the following mechanisms:

- decrypt encryption decipherment keys using asymmetric mechanisms;
- digital authentication : authentication of the TOE (on behalf of the TOE holder) using an asymmetric private key;

Moreover, the TOE ensures these keys remain genuine by enforcing an access control over the update of these keys, in order to ensure that only entitled entities can change them.

[1] It is a non-qualified advanced electronic signature if it is based in a non-qualified certificate for the SVD

5.3.2.4 P.EAC_Terminal

Terminals that intent to be Authentication Terminals must implement the respective terminal part of the protocols required to execute EAC protocol, and store (static keys) or generate (temporary keys and nonces) the corresponding credentials.

5.3.2.5 P.Terminal_PKI

The electronic document issuer shall establish a public key infrastructure for the card verifiable certificates used for Terminal Authentication. For this aim, the electronic document issuer shall run a Country Verifying Certification Authority. The instances of the PKI shall fulfill the requirements and rules of the corresponding certificate policy. The electronic document issuer shall make the CVCA certificate available to the personalization agent or the manufacturer.

5.3.2.6 P.Card_PKI PKI for sealing data in the electronic document

The description below states the responsibilities of involved parties and represents the logical, but not the physical structure of the PKI. Physical distribution ways shall be implemented by the involved parties in such a way that all certificates belonging to the PKI are securely distributed / made available to their final destination, e.g. by using directory services.

1) The electronic document Issuer shall establish a public key infrastructure to ensure the integrity and authenticity of the content of the electronic document, through the generation of digital seals protecting the data it contains. For this aim, it runs an Issuer CA.
2) The Issuer CA shall securely generate, store and use the Issuer CA key pair. The Issuer CA shall keep the Issuer CA Private Key secret.

5.3.2.7 P.Pre-Operational

1) The electronic document Issuer issues the electronic document and approves it using the terminals complying with all applicable laws and regulations.
2) The electronic l document Issuer guarantees correctness of the user data (amongst other of those, concerning the travel document holder) and of the TSF-data permanently stored in the TOE.
3) The electronic document Issuer uses only such TOE's technical components (IC) which enable traceability of the electronic documents in their manufacturing and issuing life cycle phases, i.e. before they are in the operational phase,
4) If the electronic document Issuer authorizes a Personalization Agent to personalize the electronic document for electronic document holders, the electronic document Issuer has to ensure that the Personalization Agent acts in accordance with the electronic document Issuer's policy.

5.3.2.8 P.Terminal

The BAT shall operate their terminals as follows:

1) The related terminals shall be used by terminal operators and by electronic document holders.
2) They shall implement the terminal parts of the PACE protocol and check the digital seal generated by the Issuer CA included in the electronic document and protecting its content. The BAT shall use randomly and (almost) uniformly selected nonces, if required by the protocols (for generating ephemeral keys for Diffie-Hellmann).
3) They shall also securely store the Issuer CA certificate in order to be able to verify the digital seals generated by the Issuer CA and included in the electronic document to protect its content (integrity and authenticity of the data stored in the electronic document).
4) The related terminals and their environment shall ensure confidentiality and integrity of respective data handled by them (e.g. confidentiality of PACE passwords or the PIN, integrity of PKI certificates, etc.).

5.3.2.9 P.Trustworthy_PKI

The Issuer CA shall ensure that it issues its certificates exclusively to the rightful organizations and that they create exclusively correct digital seals to be stored on the electronic document.

5.4 Assumptions

5.4.1 A.CGA Trustworthy certificate generation application

The CGA protects the authenticity of the signatory's name or pseudonym and the SVD in the (qualified) certificate by an advanced electronic signature of the CSP.

5.4.2 A.SCA Trustworthy signature creation application

The signatory uses only a trustworthy SCA. The SCA generates and sends the DTBS/R of the data the signatory wishes to sign in a form appropriate for signing by the TOE.

5.4.3 A.CSP Secure SCD/SVD management by SCD

The CSP uses only a trustworthy SCD/SVD generation device and ensures that this device can be used by authorized user only. The CSP ensures that the SCD generated practically occurs only once, that generated SCD and SVD actually correspond to each other and that SCD cannot be derived from the SVD. The CSP ensures the confidentiality of the SCD during generation and export to the TOE, does not use the SCD for creation of any signature and irreversibly deletes the SCD in the operational environment after export to the TOE.

6 Security Objectives

6.1 Security Objectives for the TOE

6.1.1 Security Objectives drawn from the protection profiles

6.1.1.1 OT.Lifecycle_Security *Lifecycle security*

The TOE shall detect flaws during the initialization, Personalization and operational usage. The TOE shall securely destroy the SCD on demand of the signatory.

6.1.1.2 OT.SCD/SVD_Auth_Gen *Authorized SCD/SVD generation*

The TOE shall provide security features to ensure that authorized users only may invoke the generation of the SCD and the SVD.

6.1.1.3 OT.SCD_Unique *Uniqueness of the signature creation data*

The TOE shall ensure the cryptographic quality of an SCD/SVD pair it creates as suitable for the advanced or qualified electronic signature. The SCD used for signature creation shall practically occur only once and shall not be reconstructable from the SVD. In that context 'practically occur once' means that the probability of equal SCDs is negligible.

6.1.1.4 OT.SCD_SVD_Corresp *Correspondence between SVD and SCD*

The TOE shall ensure the correspondence between the SVD and the SCD generated by the TOE. This includes unambiguous reference of a created SVD/SCD pair for export of the SVD and in creating an electronic signature creation with the SCD.

6.1.1.5 OT.SCD_Auth_Imp *Authorized SCD import*

The TOE shall provide security features to ensure that authorized users only may invoke the import of the SCD

6.1.1.6 OT.SCD_Secrecy *Secrecy of the signature creation data*

The secrecy of the SCD (used for signature creation) shall be reasonably assured against attacks with a high attack potential.

6.1.1.7 OT.Sig_Secure *Cryptographic security of the electronic signature*

The TOE shall create digital signatures that cannot be forged without knowledge of the SCD through robust encryption techniques. The SCD shall not be reconstructable using the digital signatures or any other data exportable from the TOE. The digital signatures shall be resistant against these attacks, even when executed with a high attack potential.

6.1.1.8 OT.Sigy_SigF *Signature creation function for the legitimate signatory only*

The TOE shall provide the digital signature creation function for the legitimate signatory only and protects the SCD against the use of others. The TOE shall resist attacks with high attack potential.

6.1.1.9 OT.DTBS_Integrity_TOE *DTBS/R integrity inside the TOE*

The TOE must not alter the DTBS/R. As by definition of the DTBS/R this may consist of the DTBS themselves, this objective does not conflict with a signature creation process where the TOE hashes the provided DTBS (in part or entirely) for signature creation.

6.1.1.10 OT.EMSEC_Design *Provide physical emanations security*

The TOE shall be designed and built in such a way as to control the production of intelligible emanations within specified limits.

6.1.1.11 OT.Tamper_ID *Tamper detection*

The TOE shall provide system features that detect physical tampering of its components, and uses those features to limit security breaches.

6.1.1.12 OT.Tamper_Resistance *Tamper resistance*

The TOE shall prevent or resist physical tampering with specified system devices and components.

6.1.1.13 OT.TOE_SSCD_Auth *Authentication proof as SSCD*

The TOE shall hold unique identity and authentication data as SSCD and provide security mechanisms to identify and to authenticate itself as SSCD.

6.1.1.14 OT.TOE_TC_SVD_Exp *TOE trusted channel for SVD export*

The TOE shall provide a trusted channel to the CGA to protect the integrity of the SVD exported to the CGA. The TOE shall enable the CGA to detect alteration of the SVD exported by the TOE.

6.1.2 Additional Security Objectives for the TOE

6.1.2.1 OT.Authentication_Secure *Secure authentication mechanisms*

The natural person can authenticate itself to the TOE via the PACE protocol and/or the VERIFY PIN command. Notice that The usage of PACE protocol is mandatory only for contactless mode.
The TOE provides (1) strong mechanism to authenticate external user/entity, and (2) strong mechanisms to authenticate the TOE.

Mechanisms to perform mutual authentication
These mechanisms aim at (1) authenticating the TOE to the outside entity, and (2) authenticating the outside entity to the TOE.
In phase 7, the mechanisms rely on asymmetric cryptography, while before phase 7 they rely on symmetric cryptography.
In the course of the mutual authentication, the TOE authenticates the outside entity using a freshly generated random number in order to avoid replay attacks.
Moreover, these mechanisms also generate trusted channel ensuring integrity, authenticity, and confidentiality of the communication using strong encryption techniques. It also ensures protection against deletion, and modification of commands.
Moreover, the TOE ensures the key its uses are genuine by enforcing an access control over the authentication keys update, in order to ensure that only entitled entities can change key values.
These mechanisms provided by the electronic document's chip are protected against attacks with high attack potential.

Mechanisms to authenticate the TOE
These mechanisms rely on asymmetric cryptography and ensure that (1) the cryptogram can not be forged without the knowledge of the authentication key, and (2) they can not be reconstructed from the authentication cryptograms.

Moreover the TOE ensures the key its uses are genuine by enforcing an access control over the authentication keys update, in order to ensure that only entitled entities can change key values.

These mechanisms provided by the electronic document's chip are protected against attacks with high attack potential.

Protection against Abuse of Functionality:
The TOE must prevent that functions of the TOE, which may not be used in TOE operational phase, can be abused in order (i) to manipulate or to disclose the User Data stored in the TOE, (ii) to manipulate or to disclose the TSF-data stored in the TOE, (iii) to manipulate (bypass, deactivate or modify) soft-coded security functionality of the TOE.

Protection against Information Leakage:
The TOE must provide protection against disclosure of confidential User Data or/and TSF-data stored and/or processed by the travel document

- by measurement and analysis of the shape and amplitude of signals or the time between events found by measuring signals on the electromagnetic field, power consumption, clock, or I/O lines,
- by forcing a malfunction of the TOE and/or
- by a physical manipulation of the TOE.

Protection against Malfunctions:
The TOE must ensure its correct operation. The TOE must prevent its operation outside the normal operating conditions where reliability and secure operation have not been proven or tested. This is to prevent functional errors in the TOE. The environmental conditions may include external energy (esp. electromagnetic) fields, voltage (on any contacts), clock frequency or temperature. The following TOE security objectives address the aspects of identified threats to be countered involving TOE's environment.

Protection against Physical Tampering:
The TOE must provide protection of confidentiality and integrity of the User Data, the TSF-data and the electronic document's Embedded Software by means of

- measuring through galvanic contacts representing a direct physical probing on the chip's surface except on pads being bonded (using standard tools for measuring voltage and current) or
- measuring not using galvanic contacts, but other types of physical interaction between electrical charges (using tools used in solid-state physics research and IC failure analysis),
- manipulation of the hardware and its security functionality, as well as
- controlled manipulation of memory contents (User Data, TSF-data)with a prior
- reverse-engineering to understand the design and its properties and functionality.

6.1.2.2 OT.Key_Lifecycle_Security *Life cycle security of the keys stored in the TOE*

The TOE shall detect flaws during the initialization, Personalization and operational usage. The TOE shall provide safe destruction techniques for the keys (other than the SCD) it stores in case of erasure, re-import or re-generation.

6.1.2.3 OT.Keys_Secrecy *Secrecy of Keys*

The secrecy of the keys (other than the SCD) stored in the TOE is reasonably assured against attacks with a high attack potential.

6.1.2.4 OT.TOE_AuthKey_Unique *Uniqueness of the TOE authentication key(s)*

The TOE shall ensure the cryptographic quality of the asymmetric authentication key pair used for the TOE authentication. The private key used for TOE authentication can practically occur only once and cannot be reconstructed from the public key. In that context 'practically occur once' means that the probability of equal TOE authentication key is negligible low.

6.1.2.5 OT.Lifecycle_Management *Management of the life cycle*

The TOE provides a life cycle management enabling to separate its life cycle in two main phases.
The first one (phase 6) is the one during which the TOE is under the sole control of the Personalization Agent. The following operation may be realized:

- The **SCD**, **SVD** and keys may be created, generated, imported or erased
- The **PINs/PUK** (s) may be created and loaded
- **SVD** and public keys may be exported

Once performed, the Personalization Agent switches the TOE in phase 7. This transition is irreversible leaving the TOE under the sole control of the R.Sigy and R.Admin according to the TOE specification and security rules set by the Personalization Agent.

6.1.2.6 OT.eServices *Provision of eServices*

The TOE provides the following mechanisms:

- decrypt encryption decipherment keys using asymmetric mechanisms;
- digital authentication : authentication of the TOE (on behalf of the TOE holder) using an asymmetric private key;

Moreover, the TOE ensures these keys remain genuine by enforcing an access control over the update of these keys, in order to ensure that only entitled entities can change them.

These mechanisms provided by the electronic document's chip are protected against attacks with high attack potential.

6.1.2.7 OT.AC_Pers_EAC *Personalization of the Electronic Document*

The TOE must ensure that user data and TSF-Data that are permanently stored in the TOE can be written by authorized personalization agents only, with the following exception: an Authentication Terminal may also write or modify user data according to its effective authorization. The effective authorization is determined by the electronic document during Terminal Authentication.

6.1.2.8 OT.Tracing *Tracing electronic document*

The TOE must prevent gathering TOE tracing data by means of unambiguous identifying the electronic document remotely through establishing or listening to a communication via the contactless interface of the TOE without knowledge of the correct values of shared passwords (PACE passwords) in advance.

6.2 Security Objectives for the Operational Environment

6.2.1 Security Objectives drawn from the protection profiles

6.2.1.1 OE.SVD_Auth *Authenticity of the SVD*

The operational environment shall ensure the integrity and authenticity of the SVD sent to the CGA of the CSP. The CGA verifies the correspondence between the SCD in the SSCD of the signatory and the SVD in the qualified certificate.

6.2.1.2 OE.CGA_QCert *Generation of qualified certificates*

The CGA shall generate a qualified certificate that includes (amongst others)

(a) the name of the signatory controlling the TOE,
(b) the SVD matching the SCD stored in the TOE and being under sole control of the signatory,
(c) the advanced signature of the CSP.

The CGA shall confirm with the generated qualified certificate that the SCD corresponding to the SVD is stored in a SSCD.

6.2.1.3 OE.Dev_Prov_Service *Authentic SSCD provided by SSCD Provisionning Service*

The SSCD provisionning service handles authentic devices that implement the TOE, prepares the TOE for proof as SSCD to external entities, personalises the TOE for the legitimate user as signatory, links the identity of the TOE as SSCD with the identity of the legitimate user, and delivers the TOE to the signatory.

6.2.1.4 OE.HID_VAD *Protection of the VAD*

If an external device provides the human interface for user authentication, this device shall ensure confidentiality and integrity of the VAD as needed by the authentication method employed from import through its human interface until import through the TOE interface. In particular, if the TOE requires a trusted channel for import of the VAD, the HID shall support usage of this trusted channel.

6.2.1.5 OE.DTBS_Intend *SCA sends data intended to be signed*

The signatory shall use a trustworthy SCA that
- generates the DTBS/R of the data that has been presented as DTBS and which the signatory intends to sign in a form which is appropriate for signing by the TOE,
- sends the DTBS/R to the TOE and enables verification of the integrity of the DTBS/R by the TOE,
- attaches the signature produced by the TOE to the data or provides it separately.

6.2.1.6 OE.DTBS_Protect *SCA protects the data intended to be signed*

The operational environment shall ensure that the DTBS/R cannot be altered in transit between the SCA and the TOE. In particular, if the TOE requires a trusted channel for import of DTBS/R, the SCA shall support usage of this trusted channel.

6.2.1.7 OE.Signatory *Security obligation of the signatory*

The signatory shall check that the SCD stored in the SSCD received from SSCD-provisioning service is in non-operational state. The signatory shall keep their VAD confidential.

6.2.1.8 OE.SCD/SVD_Auth_Gen *Authorized SCD/SVD generation*

The CSP shall provide security features to ensure that authorised users only may invoke the generation of the SCD and the SVD.

6.2.1.9 OE.SCD_Secrecy *SCD Secrecy*

The CSP shall protect the confidentiality of the SCD during the generation and export to the TOE. The CSP shall not use the SCD for creation of any signature and shall irreversibly delete the SCD in the operational environment after export to the TOE.

6.2.1.10 OE.SCD_Unique *Uniqueness of the signature creation data*

The CSP shall ensure the cryptographic quality of the SCD/SVD pair, which is generated in the environment, for the qualified or advanced electronic signature. The SCD used for signature creation shall paractically occur only once, i.e. the probability of equal SCDs shall be negligible, and the SCD shall not be reconstructable from the SVD.

6.2.1.11 OE.SCD_SVD_Corresp *Correspondance between SVD and SCD*

The CSP shall ensure the correspondance between the SVD and the SCD generated by the CSP. This includes the correspondence between the SVD send to the CGA and the SCD exported to the TOE of the signatory identified in the SVD certificate.

6.2.1.12 OE.CGA_SSCD_Auth *Pre-initialisation of the TOE for SSCD authentication*

The CSP shall check by means of the CGA whether the device presented for application of a (qualified) certificate holds unique identification as SSCD, successfully proved this identity as SSCD to the CGA, and whether this identity is linked to the legitimate holder of the device as applicant for the certificate.

6.2.1.13 OE.CGA_TC_SVD_Imp *CGA trusted channel for SVD import*

The CGA shall detect alteration of the SVD imported from the TOE with the claimed identity of the SSCD.

6.2.2 Additional security objectives for the operational environment

6.2.2.1 OE.LinkSCD_QualifiedCertificate *Link between SCD stored in the TOE and the relevant qualified certificate*

The Role in charge of creating and updating the SCD (**Personalization Agent**, **R.Admin**), or the trusted IT entity involved in the updating process (CSP**)** shall ensure an unambiguous link between the (qualified) certificate(s) and the corresponding SCD(s). This link might be figured out by a PKCS#15 structure, an XML structure, an identifier linking the file containing the (qualified) certificate or the URL hosting them to the SCD(s) stored in the TOE. In particular, it implies this link is updated, each time the SCD(s) is created, imported, erased or generated.

6.2.2.2 OE.AuthKey_Transfer *Secure transfer of authentication key(s) to the TOE*

The entity in charge of generating the authentication keys to be loaded in the TOE shall ensure the confidentiality of the key(s) transferred to the TOE.

6.2.2.3 OE.AuthKey_Unique *Uniqueness of the authentication key(s)*

The entity in charge of generating the authentication keys to be loaded in the TOE shall ensure the cryptographic quality of the authentication key(s). The authentication key used for authentication can practically occur only once and, in case of a TOE authentication key cannot be reconstructed from its public portion. In that context 'practically occur once' means that the probability of equal keys is negligible low.

6.2.2.4 OE.TOE_PublicKeyAuth_Transfer *Secure transfer of public authentication key(s) of the TOE*

The entity in charge of generating the authentication certificate from the TOE's authentication public key generated in the TOE shall ensure the authenticity of this data when transferred from the TOE. This may be achieved through operational measures.

6.2.2.5 OE.Terminal_*Authentication* *Authentication Key pairs needed for Terminal Authentication*

The electronic document issuer shall establish a public key infrastructure for the card verifiable certificates used for Terminal Authentication. For this aim, the electronic document issuer shall run a Country Verifying Certification Authority. The instances of the PKI shall fulfill the requirements and rules of the corresponding certificate policy. The electronic document issuer shall make the CVCA certificate available to the personalization agent or the manufacturer.

6.2.2.6 OE.Legislative_Compliance *Issuing of the electronic document*

The electronic document Issuer must issue the electronic document and approve it using the terminals complying with all applicable laws and regulations.

6.2.2.7 OE.Passive_Auth_Sign *Authentication of electronic document by Signature*

The electronic document Issuer has to establish the necessary public key infrastructure as follows: the Issuer CA acting on behalf and according to the policy of the electronic document Issuer must (i) generate a cryptographically secure Issuer CA Key Pair, (ii) ensure the secrecy of the Issuer CA Private Key and issue certificate for its Sub-CA in a secure operational environment (if needed), and (iii) publish its certificate. Hereby authenticity and integrity of these certificates are being maintained.
An Issuer CA acting in accordance with the Issuer CA policy must generate digital seals protecting the content of electronic document in a secure operational environment only.
The Issuer CA must issue its certificates exclusively to the rightful organizations (and must sign exclusively correct digital seals to be stored on electronic document).

6.2.2.8 OE.Personalization Personalization of *electronic* document

The *electronic* document Issuer must ensure that the Personalization Agents acting on his behalf (i) store the corresponding data in the electronic document (electronic Personalization) for the electronic document holder, (ii) write the document details data, (iii) write the initial TSF data, (iv) sign the digital seal protecting the content of the electronic document.

.

6.2.2.9 OE.Terminal *Terminal operating*

The terminal operators must operate their terminals as follows:

- The related terminals are used by terminal operators and by electronic document holders.
- The related terminals implement the terminal parts of the PACE protocol and check the digital seal generated by the Issuer CA included in the electronic document and protecting its content. The BAT shall use randomly and (almost) uniformly selected nonces, if required by the protocols (for generating ephemeral keys for Diffie-Hellmann).
- The related terminals securely store the Issuer CA certificate in order to be able to verify the digital seals generated by the Issuer CA and included in the electronic document to protect its content (integrity and authenticity of the data stored in the electronic document). The Authentication Terminal will use certificates issued by the eService certification authority(ies) to execute the EAC protocol and perform terminal authentication.
- The related terminals and their environment must ensure confidentiality and integrity of respective data handled by them (e.g. confidentiality of the PACE passwords or the PIN, integrity of PKI certificates, etc.), where it is necessary for a secure operation of the TOE.

6.2.2.10 OE.Electronic _Document_Holder Electronic *document holder Obligations*

The electronic document holder may reveal, if necessary, his or her verification values of the PACE password or the PIN to an authorized person or device who definitely act according to respective regulations and are trustworthy.

6.3 Security Objectives Rationale

6.3.1 Security objectives/Threats backtracking

	T.SCD_Divulg	T.SCD_Derive	T.Hack_Phys	T.SVD_Forgery	T.SigF_Misuse	T.DTBS_Forgery	T.Sig_Forgery	T.Key_Divulg	T.Key_Derive	T.TOE_PublicAuthKey_Forgery	T.Authentication_Replay	T.Forgery	T.Counterfeit	T.Sensitive_Data	T.Eavesdropping	T.Skimming	T.Abuse-Func	T.Information_Leakage	T.Malfunction	T.Phys-Tamper	T.Tracing
OT.Lifecycle_Security					X																
OT.SCD/SVD_Auth_Gen		X																			
OT.SCD_Unique							X														
OT.SCD_SVD_Corresp				X																	
OT.SCD_Auth_Imp	X																				
OT.SCD_Secrecy	X		X																		
OT.Sig_Secure		X					X														
OT.Sigy_SigF					X																
OT.DTBS_Integrity_TOE					X	X															
OT.EMSEC_Design			X																		
OT_Tamper_ID			X																		
OT_Tamper_Resistance			X																		
OT_TOE_SSCD_Auth																					
OT.TOE_TC_SVD_Exp				X																	
OT.Authentication_Secure									X		X	X	X	X	X	X	X	X	X	X	
OT.Key_Lifecycle_Security								X													
OT.Keys_Secrecy			X					X													
OT.TOE_AuthKey_Unique									X												
OT.Lifecycle_Management					X																
OT.eServices																					
OT.AC_Pers_EAC												X									
OT.Tracing																					X
OE.SVD_Auth				X																	
OE.CGA_QCert							X														
OE.Dev_Prov_Service																					
OE.HID_VAD					X																
OE.DTBS_Intend					X	X															
OE.DTBS_Protect					X	X															
OE.Signatory					X																
OE.SCD/SVD_Auth_Gen	X																				
OE.SCD_Secrecy	X																				
OE.SCD_Unique		X					X														
OE.SCD_SVD_Corresp				X																	
OE.CGA_SSCD_Auth																					
OE.CGA_TC_SDV_Imp				X																	
OE.LinkSCD_QualifiedCertificate																					
OE.AuthKey_Transfer								X					X								
OE.AuthKey_Unique									X				X								
OE.TOE_PublicKeyAuth_Transfer										X			X								
OE.Terminal_Authentication														X		X					
OE.Legislative_Compliance																					
OE.Passive_Auth_Sign												X									
OE.Personalization												X									
OE.Terminal												X									
OE.Electronic_Document_Holder																X					X

6.3.2 Security objectives/OSPs backtracking

	P.CSP_QCert	P.QSign	P.Sigy_SSCD	P.Sig_Non-Repud	P.LinkSCD_QualifiedCertificate	P.TOE_PublicAuthKey_Cert	P.eServices	P.Pre-Operational	P.Terminal	P.EAC_Terminal	P.Terminal_PKI	P.Card_PKI	P.Trustworthy_PKI	A.CGA	A.SCA	A.CSP
OT.Lifecycle_Security	X		X	X												
OT.SCD/SVD_Auth_Gen			X													
OT.SCD_Unique			X	X												
OT.SCD_SVD_Corresp	X			X												
OT.SCD_Auth_Imp	X		X													
OT.SCD_Secrecy			X	X												
OT.Sig_Secure		X	X	X												
OT.Sigy_SigF		X	X	X												
OT.DTBS_Integrity_TOE			X	X												
OT.EMSEC_Design			X	X												
OT_Tamper_ID				X												
OT_Tamper_Resistance			X	X												
OT_TOE_SSCD_Auth	X		X	X												
OT.TOE_TC_SVD_Exp			X	X												
OT.Authentication_Secure																
OT.Key_Lifecycle_Security																
OT.Keys_Secrecy																
OT.TOE_AuthKey_Unique																
OT.Lifecycle_Management																
OT.eServices							X									
OT.AC_Pers_EAC								X								
OT.Tracing																
OE.SVD_Auth				X										X		
OE.CGA_QCert	X	X		X										X		
OE.Dev_Prov_Service			X	X												
OE.HID_VAD																
OE.DTBS_Intend		X		X											X	
OE.DTBS_Protect				X												
OE.Signatory				X												
OE.SCD/SVD_Auth_Gen	X		X	X												X
OE.SCD_Secrecy			X	X												X
OE.SCD_Unique			X	X												X
OE.SCD_SVD_Corresp	X			X												X
OE.CGA_SSCD_Auth	X		X	X												
OE.CGA_TC_SDV_Imp			X	X												
OE.LinkSCD_QualifiedCertificate				X	X											
OE.AuthKey_Transfer										X						
OE.AuthKey_Unique										X						
OE.TOE_PublicKeyAuth_Transfer						X				X						
OE.Terminal_Authentication										X	X					
OE.Legislative_Compliance								X								
OE.Passive_Auth_Sign												X	X			
OE.Personalization								X								
OE.Terminal									X	X						
OE.Electronic_Document_Holder																

6.3.3 Security objectives sufficiency

T.SCD_Divulg (*storing,copying and releasing of the signature creation data*) adresses the threat against the legal validity of electronic signature due to storage and copying of SCD outside the TOE, as expressed in recital (18) of the directive. This threat is countered by:

- **OT.SCD_Secrecy**, which assures the secrecy of the SCD used by the TOE for signature creation
- **OE.SCD_Secrecy**, which assures the secrecy of the SCD in the CSP environment

Furthermore, generation and/or import of SCD known by an attacker is countered by **OE.SCD/SVD_Auth_Gen**, which ensures that only authorized SCD generation in the environment is possible, and **OT.SCD_Auth_Imp**, which ensures that only SCD import is possible.

T.SCD_Derive (*Derive the signature creation data*) deals with the attacks on the SCD via public known data produced by the TOE, which are the SVD and the signatures created with the SCD. This threat is countered by:

- **OT.SCD/SVD_Auth_Gen** by implementing cryptographically secure generation of the SCD/SVD pair.
- **OT.Sig_Secure**, which ensures cryptographically secure electronic signature.
- **OE.SCD_Unique** by implementing cryptographically secure generation of the SCD/SVD pair

T.Hack_Phys (*Exploitation of physical vulnerabilities*) deals with physical attacks exploiting physical vulnerabilities of the TOE. **OT.SCD_Secrecy** preserves the secrecy of the SCD. **OT_EMSEC_Design** counters physical attacks through the TOE interfaces and observation of TOE emanations. **OT.Tamper_ID** and **OT.Tamper_Resistance** counter the threat T.Hack_Phys by detecting and resisting tampering attacks.
OT.Keys_Secrecy preserves the secrecy of all the authentication and eServices keys stored in the TOE.

T.SVD_Forgery (*Forgery of the signature verification data*) deals with the forgery of the SVD given to the CGA for certificate generation. T.SVD_Forgery is addressed by

- **OT.SCD_SVD_Corresp**, which ensures correspondence between SCD and SVD and unambiguous reference of the SCD/SVD pair for the SVD export and signature creation with the SCD
- **OE.SCD_SVD_Corresp**, wich ensures correspondence between SVD and SCD
- **OE.SVD_Auth** that ensures the integrity of the SVD given to the CGA of the CSP and verification of the correspondence between the SCD in the SSCD of the signatory and the SVD in the input it provides to the certificate generation function of the CSP.
- **OT.TOE_TC_SVD_Exp**, which ensures that the TOE sends the SVD in a verifiable form through a trusted channel to the CGA
- **OE.CGA_TC_SVD_Imp**, which provides verification of SVD authenticity by the CGA

T.SigF_Misuse (*Misuse of the signature creation function of the TOE*) addreses the threat of misuse of the TOE signature creation function by other than the signatory to create an electonic signature on data which the signatory has not expressed the intent to sign, as required by paragraph 1(c) of Annex III. **OT.Lifecycle_Security** (*Lifecycle security*) requires the TOE to detect flaws during initialisation, Personalization and operational usage including secure destruction of the SCD, which may be initiated by the signatory. **OT_Sigy_SigF** (*Signature creation function for the legitimate signatory only*) ensures that the TOE provides the signature creation function for the legitimate signatory only. **OE_DTBS_Intend** (*Data intended to be signed*) ensures that the SCA sends the DTBS/R only for data the signatory intends to sign and **OE.DTBS_Protect** counters manipulation of the DTBS during during transmission over the channel between the SCA and the TOE. **OT.DTBS_Integrity_TOE** (*DTBS/R integrity inside the TOE*) prevents the DTBS/R from alteration inside the TOE. If the SCA provides a human interface for user authentication. **OE.HID_VAD** (*protection of the VAD*) provides confidentiality and integrity of the VAD as needed by the authentication method employed. **OE.Signatory** ensures that the signatory checks that an SCD stored in the SSCD when received from an SSCD-Provisioning service provider is in non-operational state, i.e. the SCD cannot be used before the signatory becomes control over the SSCD. OE.signatory ensures also that the signatory keeps their VAD confidential.
OT.LifeCycle_Management ensures that when the TOE is under the Personalization Agent control, it cannot be misused to sign on behalf of the legitimate Signatory.

T.DTBS_Forgery (*Forgery of the DTBS/R*) addresses the threat arising from modifications of the DTBS/R sent as input to the TOE's signature creation function that does not represent the DTBS as presented to the signatory and for which the

signature has expressed its intent to sign intends to sign. The TOE IT environment addresses **T.DTBS_Forgery** by the means of **OE.DTBS_Intend**, which ensures that the trustworthy SCA generates the DTBS/R of the data that has been presented as DTBS and which the signatory intends to sign in a form appropriate for signing by the TOE, and by means of **OE.DTBS_Protect**, which ensures that the DTBS/R cannot be altered in transit between the SCA and the TOE. The TOE counters this threat by the means of **OT.DTBS_Integrity_TOE** by ensuring the integrity of the DTBS/R inside the TOE.

T.Sig_Forgery (*Forgery of the electronic signature*) deals with non-detectable forgery of the electronic signature. **OT.Sig_Secure**, **OT.SCD_Unique**, **OE.SCD_Unique** and **OE.CGA_QCert** address this threat in general. OT.Sig_Secure (*Cryptographic security of the electronic signature*) ensures by means of robust cryptographic techniques that the signed data and the electronic signature are securely linked together. OT.SCD_Unique and OE.SCD_Unique ensure that the same SCD cannot be generated more than once and the corresponding SVD cannot be included in another certificate by chance. OE.CGA_QCert prevents forgery of the certificate for the corresponding SVD, which would result in false verification decision concerning a forged signature.

T.Key_Divulg addresses the threat against the (1) authentication key of the TOE, (2) the authentication keys of entities and (3) the eServices keys stored in the TOE due to storage and copying of key(s) outside the TOE. This threat is countered by **OT.Keys_Secrecy** which assures the secrecy of the keys stored and used by the TOE. **OE.AuthKey_Transfer** ensures the confidentiality of the authentication keys transferred to the TOE.
OT.Key_Lifecycle_Security (*Lifecycle security*) ensures the secrecy of the keys stored in the TOE during the whole life of the TOE.

T.Key_Derive deals with attacks on authentication and eServices keys via public known data produced or received by the TOE (public key, authentication cryptogram,…). This threat is countered by **OE.AuthKey_Unique** (in case of import) and **OT.TOE_AuthKey_Unique** (in case of TOE's authentication key generation) that provides cryptographic secure generation of the keys. **OT.Authentication_Secure** ensures secure authentication cryptograms.

T.TOE_PublicAuthKey_Forgery deals with the forgery of the TOE's public key used for authentication exported by the TOE to an entitled entity for the generation of the certificate. This is addressed by **OE.TOE_PublicKeyAuth_Transfer** which ensures the authenticity of the TOE's public key for authentication.

T.Authentication_Replay deals with the threats when an attacker retrieves an authentication cryptogram presented to the TOE by an entity and presents it again to the TOE in order to grant some rights and gain access to some data on the TOE. This threat is addressed by **OT.Authentication_Secure** that ensures the authentication cryptogram can not be replayed as they rely on random data internally generated by the TOE.

T.Counterfeit addresses the attack of an unauthorized copy or reproduction of the genuine electronic document. This attack is countered by the proof of the chip's authenticity, as aimed by **OT.Authentication_Secure** using a Chip Authentication key pair that is generated within the issuing PKI branch, as aimed by **OE.AuthKey_Transfer, OE.AuthKey_Unique, OE.TOE_PublicKeyAuth_Transfer.**

T.Sensitive_Data is countered by the TOE-Objective **OT.Authentication_Secure**, that requires that read access to sensitive user data is only granted to Authentication Terminals with corresponding access rights. Furthermore, it is required that the confidentiality of the data is ensured during contactless transmission. The objective **OE.Terminal_Authentication** requires the electronic document issuer to provide the public key infrastructure (PKI) to generate and distribute the card verifiable certificates needed by the electronic document to securely authenticate the Authentication Terminal.

T.Abuse-Func addresses attacks of misusing TOE's functionality to manipulate or to disclosure the stored User- or TSF-data as well as to disable or to bypass the soft-coded security functionality. The security objective **OT.Authentication_Secure** ensures that the usage of functions having not to be used in the operational phase is effectively prevented.

T.Eavesdropping addresses listening to the contactless communication between the TOE and a BAT or an Authentication Terminal in order to gain access to transferred user data. This threat is countered by the security objective **OT.Authentication_Secure** through a trusted channel based on PACE or EAC Authentication.

T.Forgery addresses the fraudulent, complete or partial alteration of user data and/or TSF-Data stored on the TOE, and/or exchanged between the TOE and the terminal. The threat T.Forgery addresses the fraudulent, complete or partial alteration of the User Data or/and TSF-data stored on the TOE or/and exchanged between the TOE and the terminal. The security objective **OT.AC_Pers** requires the TOE to limit the write access for the travel document to the trustworthy Personalization

Agent (cf. **OE.Personalization**). The TOE will protect the integrity and authenticity of the stored and exchanged User Data or/and TSF-data as aimed by the security objective **OT.Authentication_Secure**. This objective contribute also to protecting integrity of the User Data or/and TSF-data stored on the TOE. A terminal operator operating his terminals according to **OE.Terminal** and performing the digital seal verification as aimed by **OE.Passive_Auth_Sign** will be able to effectively verify integrity and authenticity of the data received from the TOE.

T.Information_Leakage is typical for integrated circuits like smart cards under direct attack with high attack potential. The protection of the TOE against this threat is addressed by the directly related security objective **OT.Authentication_Secure**.

T.Malfunction is typical for integrated circuits like smart cards under direct attack with high attack potential. The protection of the TOE against this threat is addressed by the directly related security objectives **OT.Authentication_Secure**.

T.Phys-Tamper is typical for integrated circuits like smart cards under direct attack with high attack potential. The protection of the TOE against this threat is addressed by the directly related security objective **OT.Authentication_Secure**.

T.Skimming addresses accessing the user data (stored on the TOE or transferred between the TOE and the terminal) using the TOE's contactless interface. This threat is countered by the security objective **OT.Authentication_Secure** through the PACE authentication. The objective **OE.Electronic_Document_Holder** ensures that a PACE session can only be established either by the travel document holder itself or by an authorised person or device, and, hence, cannot be captured by an attacker. Additionally, the threat is also addressed by **OT.Authentication_Secure** that demands a trusted channel based on Chip Authentication, and requires that read access to sensitive user data is only granted to Authentication Terminals with corresponding access rights. Moreover, **OE.Terminal_Authentication** requires the electronic document issuer to provide the corresponding PKI.

T.Tracing addresses gathering TOE tracing data identifying it remotely by establishing or listening to a communication via the contactless interface of the TOE, whereby the attacker does not a priori know the correct values of the PACE password. This threat is directly countered by security objectives **OT.Tracing** (no gathering TOE tracing data) and **OE.Electronic_Document_Holder** (the attacker does not a priori know the correct values of the shared passwords).

Enforcement of OSPs by security objectives

P.CSP_QCert (*CSP generates qualified certificates*) establishes the CSP generating qualified certificate or non-qualified certificate linking the signatory and the SVD implemented in the SSCD under sole control of this signatory. P.CSP_QCert is addressed by

- **OT.Lifecycle_Security**, which requires the TOE to detect flaws during the initialisation, Personalization and operational usage,
- **OT.SCD_SVD_Corresp**, which requires to ensure the correspondance between the SVD and the SCD during their generation, and ensures that the SVD exported by the TOE to the CGA corresponds to the SCD stored in the TOE and used by the signatory.
- **OE.CGA_QCert** for generation of qualified certificates or non-qualified certificates which requires the CGA to certify the SVD matching the SCD implemented in the TOE under sole control of the signatory.
- **OE.SCD/SVD_Auth_Gen**, which ensures that the SCD/SVD generation can be invoked by authorized users only,
- **OT.SCD_Auth_Imp** which ensures that authorised users only may invoke the import of the SCD,
- **OE.SCD_SVD_Corresp**, which requires the CSP to ensure the correspondence between the SVD and the SCD during their generation,
- **OT.TOE_SSCD_Auth**, which ensures that the copies of the TOE will hold unique identity and authentication data as SSCD and provide security mechanisms enabling the CGA to identify and to authenticate the TOE as SSCD to prove this identity as SSCD to the CGA,
- **OE.CGA_SSCD_Auth**, ensures that the SP checks the proof of the device presented of the applicant that it is a SSCD.

P.QSign (*Qualified electronic signatures*) provides that the TOE and the SCA may be employed to sign data with an advanced electronic signature, which is a qualified electronic signature if based on a valid qualified certificate. **OT.Sigy_SigF** ensures signatory's sole control of the SCD by requiring the TOE to provide the signature creation function for the legitimate signatory only and to protect the SCD against the use of others. **OT.Sig_Secure** ensures that the TOE creates electronic signatures, which cannot be forged without knowledge of the SCD through robust encryption techniques.

OE.CGA_QCert addresses the requirement of qualified or non-qualified electronic certificates building a base for the electronic signature. **OE.DTBS_Intend** ensures that the SCA provides only those DTBS to the TOE, which the signatory intends to sign.

P.Sigy_SSCD *(TOE as secure signature* creation device) requires the TOE to meet Annex III. This is ensured as follows:

- **OT.SCD_Unique** and **OE.SCD_Unique** meet the paragraph 1(a) of Annex III, by the requirements that the SCD used for signature creation can practically occur only once;
- **OT.SCD_Unique**, **OE.SCD_Unique**, **OT.SCD_Secrecy** and **OT.Sig_Secure** meet the requirement in paragraph 1(a) of Annex III by the requirements to ensure secrecy of the SCD. **OT.EMSEC_Design** and **OT.Tamper_Resistance** address specific objectives to ensure secrecy of the SCD against specific attacks;
- **OT.SCD_Secrecy** and **OT.Sig_Secure** meet the requirement in paragraph 1(b) of Annex III by the requirements to ensure that the SCD cannot be derived from SVD, the electronic signatures or any other data exported outside the TOE;
- **OT.Sigy_SigF** and **OE.SCD_Secrecy** meet the requirement in paragraph 1(c) of Annex III by the requirements to ensure that the TOE provides the signature creation function for the legitimate signatory only and protects the SCD against the use of others;
- **OT.DTBS_Integrity_TOE** meets the requirements in paragraph 2 of Annex III as the TOE must not alter the DTBS/R.

Paragraph 2 of Annex III, requires that an SSCD does not prevent the data to be signed from being presented to the signatory prior to the signature process is obviously fulfilled by the method of TOE usage: the SCA will present the DTBS to the signatory and send it to the SSCD for signing. The usage of SCD under sole control of the signatory is ensured by

- **OT.Lifecycle_Security** requiring the TOE to detect flaws during the initialisation, Personalization and operational usage,
- **OT.SCD/SVD_Auth_Gen** and **OE.SCD/SVD_Auth_Gen** which limit invocation of the generation of the SCD and the SVD to authorized users only,
- **OT.SCD_Auth_Imp**, which limits the SCD import to authorised users only,
- **OE.SCD_Secrecy**, which ensures the confidentiality of the SCD during generation and export to the TOE, and deletes the SCD after export to the TOE. The CSP does not use the SCD for signature creation,
- **OT.Sigy_SigF**, which requires the TOE to provide the signature creation function for the legitimate signatory only and to protect the SCD against the use of others.

OE.Dev_Prov_Service ensures that the legitimate user obtains a TOE sample as an authentic, initialized and personalised TOE from an SSCD Provisioning Service through the TOE delivery procedure. If the TOE implements SCD generated under control of the SSCD Provisionning Service the legitimate user receives the TOE as SSCD. If the TOE delivered to the legitimate user without SCD in the operational phase he or she applies for the (qualified) certificate as the Device holder and legitimate user of the TOE. The CSP will use the TOE security feature (addressed by the security objectives **OT.TOE_SSCD_Auth** and **OT.TOE_TC_SVD_Exp**) to check whether the device presented is a SSCD linked to the applicant as required by **OE.CGA_SSCD_Auth** and the received SVD is sent by this SSCD as required by **OE.CGA_TC_SVD_Imp**. Thus the obligation of the SSCD provision service for the first SCD/SVD pair is complemented in an appropriate way by the CSP for the SCD/SVD pair generated outside the secure preparation environment.

P.Sig_Non-Repud ***(****Non-repudiation of signatures***)** deals with the repudiation of signed data by the signatory, although the electronic signature is successfully verified with the SVD contained in their certificate valid at the time of signature creation. This policy is implemented by the combination of the security objectives for the TOE and its operational environment, which ensures the aspects of signatory's sole control over and responsibility for the electronic signatures created with the TOE.
OE.Dev_Prov_Service ensures that the signatory obtains uses an authentic TOE, initialized and personalized for the signatory.
OE.SCD/SVD_Auth_Gen, OE.SCD_Secrecy and **OE.SCD_Unique** ensure the security of the SCD in the CPS environment**. OE.SCD_Secrecy** ensures the confidentiality of the SCD during generation, during and after export to the TOE. The CSP does not use the SCD for creation of any signature and deletes the SCD irreversibly after export to the TOE.
OE.SCD_Unique provides that the signatory's SCD can practically occur once .**OE_SCD_SVD_Corresp** ensures that the certificate of the signatory corresponds to the SCD that is implemented in the copy of the TOE of the signatory.
OE.CGA_QCert ensures that the certificate allows to identify the signatory and thus to link the SVD to the signatory. **OE.SVD_Auth** and **OE.CGA_QCert** require the environment to ensure authenticity of the SVD as being exported by the TOE and used under sole control of the signatory.

OT.SCD_SVD_Corresp ensures that the SVD exported by the TOE corresponds to the SCD that is implemented in the TOE.**OT.SCD_Unique** provides that the signatory's SCD can practically occur just once.
OE.Signatory ensures that the signatory checks that the SCD, stored in the SSCD received from an SSCD- provisioning service is in non-operational state (i.e. the SCD cannot be used before the signatory becomes into sole control over the SSCD). The TOE security feature addressed by the security objectives **OT.TOE_SSCD_Auth** and **OT.TOE_TC_SVD_Exp** supported by **OE.Dev_Prov_Service** enables the verification whether the device presented by the applicant is a SSCD as required by **OE.CGA_SSCD_Auth** and the received SVD sent by the device holding the corresponding SCD as required by **OE.CGA_TC_SVD_Imp**.
OT.Sigy_SigF provides that only the signatory may use the TOE for signature creation. As prerequisite **OE.Signatory** ensures that the signatory keeps their VAD confidential.
OE.DTBS_Intend, **OE.DTBS_Protect**, and **OT.DTBS_Integrity_TOE**, ensure that the TOE creates electronic signatures only for those DTBS/R, which the signatory has decided to sign as DTBS. The robust cryptographic techniques required by **OT.Sig_Secure** ensure that only this SCD may create a valid electronic signature that can be successfully verified with the corresponding SVD used for signature verification. The security objective for the TOE **OT.Lifecycle_Security** (Lifecycle security), **OT.SCD_Secrecy** (Secrecy of the signature creation data), **OT.EMSEC_Design** (Provide physical emanations security), **OT.Tamper_ID** (Tamper detection) and **OT.Tamper_Resistance** (Tamper resistance) protect the SCD against any compromise.
OT.LifeCycle_Management ensures that when the TOE is under the Personalization Agent control, it can not be misused to sign on behalf of the legitimate Signatory.
OE.LinkSCD_QualifiedCertificate ensure the SCA always uses the SCD it intends to, in order to create a digital signature.
OE.LinkSCD_QualifiedCertificate ensures that the SCA can unambiguously sort out within the TOE file structure the SCD matching any (qualified) certificate it has chosen and intends to use.

P.LinkSCD_QualifiedCertificate (*Link between a SCD and its qualified certificate*) ensures that the SCA can unambiguously find within the TOE File structure the SCD matching a (qualified) certificate it has chosen to perform an electronic signature. It is addressed by **OE.LinkSCD_QualifiedCertificate** that ensures an unambiguous link between each (qualified) certificate and the matching SCD loaded in the TOE.

P.TOE_PublicAuthKey_Cert (*Certificate for asymmetric TOE authentication keys*) ensures that each private key(s) of the TOE for authentication matches the public key stored within the relevant certificate issued by an entitled entity. The authentication public key is exported thanks to **OE.TOE_PublicKeyAuth_Transfer**.

P.eServices (*Provision of eServices*) ensures that the TOE provides secure eServices functionalities. It is addressed by **OT.eServices**.

P.EAC_Terminal addresses the requirement for Authentication Terminals to implement the terminal parts of the protocols needed to executed EAC according to its specification in [TR_03110], and to store (static keys) or generate (temporary keys and nonces) the needed related credentials. This is enforced by **OE.Chip_Auth_Key** which requires Chip Authentication and Restricted Identity keys to be correctly generated and stored, by **OE.Terminal_Authentication** for the PKI needed for Terminal Authentication, and by **OE.Terminal** which covers the PACE protocol and the digital seal verification.

P.Terminal_PKI is enforced by establishing the receiving PKI branch as aimed by the objective **OE.Terminal_Authentication**.

P.Card_PKI is enforced by establishing the issuing PKI branch as aimed by the objective **OE.Passive_Auth_Sign** (for the digital seal verification).

P.Pre-Operational is enforced by the following security objectives: (i) **OT.AC_Pers** and **OE.Personalization** together enforce the OSP's properties 'correctness of the User- and the TSF-data stored' and 'authorisation of Personalization Agents'; (ii) **OE.Legislative_Compliance** is affine to the OSP's property 'compliance with laws and regulations'.

P.Terminal is obviously enforced by the objective OE.Terminal, whereby the one-to-one mapping between the related properties is applicable.

P.Trustworthy_PKI is enforced by **OE.Passive_Auth_Sign** (for Issuer CA, issuing PKI branch).

Upkeep of assumptions by security objectives:

A.CGA (*Trustworthy certificate generation application*) establishes the protection of the authenticity of the signatory's name and the SVD in the qualified certificate by the advanced signature of the CSP by means of the CGA. This is addressed by **OE.CGA_QCert** (*Generation of qualified certificates*), which ensures the generation of qualified certificates, and by **OE.SVD_Auth** (*Authenticity of the SVD*), which ensures the protection of the integrity of the received SVD and the verification of the correspondence between the SVD and the SCD that is implemented by the SSCD of the signatory.

A.SCA (*Trustworthy signature creation application*) establishes the trustworthiness of the SCA with respect to generation of DTBS/R. This is addressed by **OE.DTBS_Intend** (*Data intended to be signed*) which ensures that the SCA generates the DTBS/R of the data that have been presented to the signatory as DTBS and which the signatory intends to sign in a form which is appropriate for being signed by the TOE.

A.CSP (*Secure SCD/SVD management by CSP*) establishes several security aspects concerning handling of SCD and SVD by the CSP. That the SCD/SVD generation device can only be used by authorized users is addressed by **OE.SCD/SVD_Auth_Gen** (*Authorized SCD/SVD generation*), that the generated SCD is unique and cannot be derived by the SVD is addressed by **OE.SCD_Unique** (*Uniqueness of the signature creation data*), that SCD and SVD correspond to each other is addressed by **OE.SCD_SVD_Corresp** (*Correspondence between SVD and SCD*), and that the SCD are kept confidential, are not used for signature generation in the environment and are deleted in the environment once exported to the TOE is addressed by **OE.SCD_Secrecy** (*SCD Secrecy*).

7 Extended components Definition

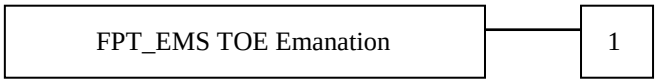
7.1 FPT_EMS TOE Emanation

The additional family FPT_EMS (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against the SCD and other secret data where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks are evaluation of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, radio emanation etc. This family describes the functional requirements for the limitation of intelligible emanations. The family FPT_EMS belongs to the Class FPT because it is the class for TSF protection. Other families within the Class FPT do not cover the TOE emanation. The definition of the family FPT_EMS is taken from the *Protection Profile Secure Signature Creation Device* [5].

Family behavior:

This family defines requirements to mitigate intelligible emanations.

Component leveling:



FPT_EMS.1 TOE Emanation has two constituents:

- FPT_EMS.1.1 Limit of Emissions requires to not emitting intelligible emissions enabling access to TSF data or user data.
- FPT_EMS.1.2 Interface Emanation requires to not emitting interface emanation enabling access to TSF data or user data.

Management:

There are no management activities foreseen.

Audit:

There are no actions identified that shall be auditable if **FAU_GEN** (*Security audit data generation*) is included in a PP or ST using FPT_EMS.1.

FPT_EMS.1 *TOE Emanation*

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_EMS.1.1 The TOE shall not emit [*assignment: types of emissions*] in excess of [*assignment: specified limits*] enabling access to [assignment*: list of types of TSF data*] and [assignment*: list of types of user data*].

FPT_EMS.1.2 The TSF shall ensure [assignment*: type of users*] are unable to use the following interface [assignment*: type of connection*] to gain access to [assignment*: list of types of TSF data*] and [assignment*: list of types of user data*].

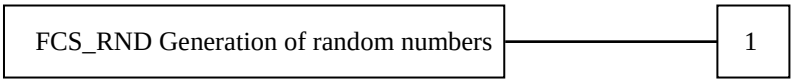
7.2 FCS_RND Generation of random numbers

To describe the IT security functional requirements of the TOE, the family FCS_RND of the class FCS (Cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes. The component FCS_RND.1 is not limited to generation of cryptographic keys unlike the component FCS_CKM.1. The similar component FIA_SOS.2 is intended for non-cryptographic use.

Family behavior:

This family defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

Component leveling:



FCS_RND.1 Generation of random numbers requires that random numbers meet a defined quality metric

Management:

There are no management activities foreseen

Audit:

There are no actions defined to be auditable

FCS_RND.1 *Quality Metric for Random Numbers*

Hierarchical to: No other components.
Dependencies: No dependencies. Definition

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: a defined quality metric].

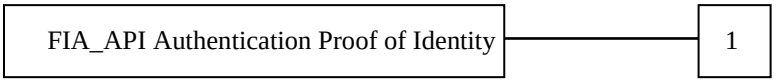
7.3 FIA_API Authentication proof of Identity

To describe the IT security functional requirements of the TOE a sensitive family (FIA_API) of the class FIA (Identification and Authentication) is defined here. This family describes the functional requirements for the proof of the claimed identity for the authentication verification by an external entity where the other families of the class FIA address the verification of the identity of an external entity.

Family behaviour:

This family defines functions provided by the TOE to prove their identity and to be verified by an external entity in the TOE IT environment.

Component leveling:



FIA_API.1 Authentication Proof of Identity.

Management:

The following actions could be considered for the management functions in FMT: Management of authentication information used to prove the claimed identity.

Audit:

There are no actions defined to be auditable.

FIA_API.1 *Authentication Proof of Identity*

Hierarchical to: No other components.
Dependencies: No dependencies. Definition

FIA.API1.1 The TSF shall provide a [assignment: authentication mechanism] to prove the identity of the [assignment: authorized user or role]

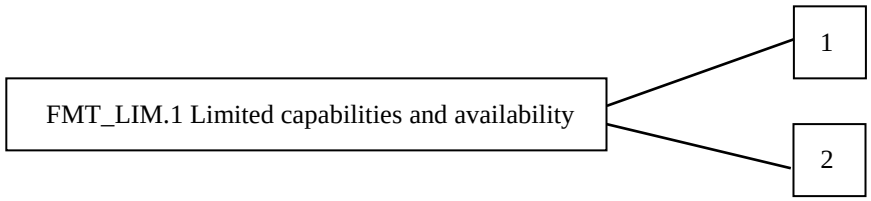
7.4 FMT_LIM Limited capabilities and availability

The family FMT_LIM describes the functional requirements for the test features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The examples of the technical mechanism used in the TOE show that no other class is appropriate to address the specific issues of preventing abuse of functions by limiting the capabilities of the functions and by limiting their availability.

Family behavior:

This family defines requirements that limit the capabilities and availability of functions in a combined manner. Note that FDP_ACF restricts the access to functions whereas the Limited capability of this family requires the functions themselves to be designed in a specific manner.

Component leveling:



FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's lifecycle.

Management:
FMT_LIM.1, FMT_LIM.2 There are no management activities foreseen.

Audit:
FMT_LIM.1, FMT_LIM.2 There are no actions defined to be auditable.

FMT_LIM.1 *Limited Capabilities*

Hierarchical to: No other components.
Dependencies: FMT_LIM.2 Limited availability

FMT_LIM.1.1 The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced [assignment: Limited capability and availability policy].

FMT_LIM.2 *Limited Availability*

Hierarchical to: No other components.
Dependencies: FMT_LIM.1 Limited capabilities

FMT_LIM.2.1 The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: Limited capability and availability policy].

8 Security Requirements

8.1 Security Functional Requirements

8.1.1 Security attributes

The security attributes and the related status for the subjects and objects are:

Subject or object the security attribute is associated with	Security Attribute type	Value of the security attribute
S.User	Role	R.Admin R.Sigy
S.User	SCD/SVD Management	Authorized Not authorized
SCD	SCD Operational	Yes No
SCD	SCD Identifier	Arbitrary value

8.1.1.1 SCD/SVD Management

In phase 6
S.Admin is the personalization agent, and as such always has the attribute "SCD/SVD Management" set to "Authorized". Furthermore in that phase, the TOE allows the SCD to be imported or generated.

In phase 7
In that phase, the TOE only supports SCD/SVD generation. The access condition for SCD/SVD generation is granted if the User is successfully authenticated as S.Admin.
If this condition is fulfilled, the attribute "SCD/SVD management" is set to "authorized", otherwise it is set to "not authorized".

8.1.1.2 SCD Operational

In phase 6
The attribute "SCD operational" is always set to "No".

In phase 7
The attribute "SCD operational" is set to "yes" as soon as the subject is authenticated as S.Signatory, using the RAD.

8.1.2 SFRs drawn for PP

The following SFRs are drawn from the protection profiles. They are sorted out depending on the life cycle of the TOE.

8.1.2.1 Phase 6&7

8.1.2.1.1 **FCS_CKM.1/SCD/SVD_Generation** Cryptographic key generation

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/SCD/SVD_Generation
The TSF shall generate an **SCD/SVD pair** in accordance with a specified cryptographic key generation algorithm:
(1) RSA key generation
(2) Key pair over Elliptic curve
and specified cryptographic key sizes:
(1) 1024 bits or 1536 bits or 2048 bits
(2) Any elliptic curve from 160 bits up to 521 bits with prime field p
that meet the following:
(1) [ANSIX9.31]
(2) [IEEE]

8.1.2.1.2 **FCS_CKM.4** *Cryptographic key destruction*

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting the buffer containing the key with zero that meets the following: none

Application note:
This SFR applies to all keys, whether it is the SCD, the SVD or another one.
The cryptographic key SCD will be destroyed before the SCD is re-imported of re-generated into the TOE.

8.1.2.1.3 **FDP_ACC.1/SCD/SVD_Generation** *Subset access control*

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attributes based access control

FDP_ACC.1.1/SCD/SVD_Generation The TSF shall enforce the SCD/SVD_Generation SFP on
(1) subjects: S.User
(2) objects: SCD, SVD
(3) operations: generation of SCD/SVD pair

8.1.2.1.4 **FDP_ACF.1/SCD/SVD_Generation** *Security attribute based access control*

Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/SCD/SVD_Generation The TSF shall enforce the SCD/SVD_Generation SFP to objects based on the following: the user S.User is associated with the security attribute "SCD/SVD management".

FDP_ACF.1.2/SCD/SVD_Generation The TSF shall enforce the following rules to determine if an operation among controlled objects is allowed: S.User with the security attribute "SCD/SVD Management" set to "authorized" is allowed to generate SCD/SVD pair.

FDP_ACF.1.3/SCD/SVD_Generation	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/SCD/SVD_Generation	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: S.User with the security attribute "SCD/SVD Management" set to "not authorized" is not allowed to generate SCD/SVD pair.

Application note:
In phase 7, the S.user can become S.admin after authentication as S.Signatory combined with an Authentication Terminal (TA_CMT).

8.1.2.1.5 **FDP_ACC.1/SVD_Transfer** *Subset access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/ SVD_Transfer	The TSF shall enforce the SVD Transfer SFP on (1) subjects: S.User, (2) objects: SVD (3) operations: export.

8.1.2.1.6 **FDP_ACF.1/SVD_Transfer** *Security attribute based access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1/ SVD_Transfer	The TSF shall enforce the SVD Transfer SFP to objects based on the following: (1) the S.User is associated with the security attribute Role, (2) the SVD.
FDP_ACF.1.2/ SVD_Transfer	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: R.Admin is allowed to export SVD.
FDP_ACF.1.3/ SVD_Transfer	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/ SVD_Transfer	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none.

Application note:
For this operation, S.User is S.Admin.
In phase 6, S.Admin is the "personalization agent" and is always allowed to export the SVD.
In phase 7, S.Admin is the subject allowed to export the SVD.

8.1.2.1.7 **FDP_ACC.1/SCD_import** *Subset access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/ SCD_Import	The TSF shall enforce the SCD_Import SFP on (1) subjects: S.User, (2) objects: SCD (3) operations: import of SCD.

8.1.2.1.8 **FDP_ACF.1/SCD_Import** *Security attribute based access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1/ SCD_Import	The TSF shall enforce the SCD_Import SFP to objects based on the following: the S.User is associated with the security attribute "SCD/SVD Management".
FDP_ACF.1.2/ SCD_Import	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: S.User with the security attribute "SCD/SVD Management" set to "authorized" is allowed to import the SCD.
FDP_ACF.1.3/ SCD_Import	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/ SCD_Import	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: S.User with security attribute "SCD/SVD Management" set to "not authorized" is not allowed to import the SCD.

Application note:
For this operation, S.User is S.Admin.
In phase 6, S.Admin is the "Personalization Agent" and always has the security attribute "SCD/SVD Management" set to "authorized".
In phase 7, SCD import is not allowed.

8.1.2.1.9 **FDP_RIP.1** *Subset residual information protection*

Hierarchical to:	No other components
Dependencies:	No dependencies
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the de-allocation of the resource from the following objects: SCD, RAD, VAD, Keys, PIN, PUK, Session keys and related data.

8.1.2.1.10 **FDP_SDI.2/Persistent** *Stored data integrity monitoring and action*

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.
FDP_SDI.2.1/ Persistent	The TSF shall monitor user data stored in containers controlled by the TSF for integrity error on all objects, based on the following attributes: integrity checked stored data.
FDP_SDI.2.2/ Persistent	Upon detection of a data integrity error, the TSF shall (1) prohibit the use of the altered data (2) inform the S.Sigy about integrity error.

Application note:
The following data persistently stored by the TOE has the user data attribute "integrity checked persistent stored data":

1. SCD
2. RAD
3. Keys

8.1.2.1.11 **FDP_ITC.1/SCD** *Import of user data without security attributes*

Hierarchical to:	No other components
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialization

FDP_ITC.1.1/SCD	The TSF shall enforce the SCD_Import SFP when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/SCD	The TSF shall ignore any security attributes associated with the **SCD** when imported from outside the TOE.
FDP_ITC.1.3/SCD	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: SCD shall be sent by an authorized CSP.

Application note:
This SFR only applies in phase 6.
The TOE interacts with a CSP through a SCD/SVD generation application to import the SCD. Authorized CSP is able to establish a trusted channel with the TOE for SCD transfer as required by FTP_ITC.1.3/SCD.
The authorized CSP is the «Personalization Agent».

8.1.2.1.12 **FDP_UCT.1/SCD** *Basic data exchange confidentiality*

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FDP_UCT.1.1/SCD	The TSF shall enforce the SCD_Import SFP to receive SCD in a manner protected from unauthorised disclosure

8.1.2.1.13 **FDP_DAU.2/SVD** Data Authentication with Identity of Guarantor

Hierarchical to:	FDP_DAU.1 Basic Data Authentication
Dependencies	FIA_UID.1 Timing if identification

FDP_DAU.2.1/SVD	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of SVD
FDP_DAU.2.2/SVD	The TSF shall provide CGA with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

8.1.2.1.14 **FIA_UID.1** *Timing of identification*

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_UID.1.1	The TSF shall allow (1) Self-test according to FPT_TST.1 (2) establishing a trusted channel between the CGA and the TOE by means of the TSF required by FTP_ITC.1/SVD

(3) establishing a trusted channel between the CSP and the TOE by means of the TSF required by FTP_ITC.1/SCD.

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.

8.1.2.1.15 **FIA_UAU.1** *Timing of authentication*

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1 The TSF shall allow
(1) Self-test according to FPT_TST.1,
(2) Identification of the user by means of TSF required by FIA_UID.1.
(3) establishing a trusted channel between the CGA and the TOE by means of the TSF required by FTP_ITC.1/SVD

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.

8.1.2.1.16 **FIA_API.1** *Authentication proof of Identity*

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide an authentication mechanism to prove the identity of the SSCD

Application note:
The authentication mechanism is achieved as follows:
In phase 6 : GP authentication
In phase 6 & 7 : an outgoing MAC

8.1.2.1.17 **FMT_SMR.1** *Security roles*

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification.

FMT_SMR.1.1 The TSF shall maintain the roles R.Admin, R.Sigy.

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

8.1.2.1.18 **FMT_SMF.1** *Security management functions*

Hierarchical to: No other components.
Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:
(1) Creation and modification of RAD,
(2) Enabling the signature creation function,

(3) Modification of the security attribute SCD/SVD management, SCD operational,
(4) Change the default value of the security attribute SCD Identifier.
(5) SCD/SVD Generation,
(6) SCD import,
(7) Unblock of RAD,
(8) Initialisation, change, resume, and unblock of PIN and PUK,
(9) Erase of PIN and RAD,
(10) Reinitialisation of PIN

8.1.2.1.19 **FMT_MSA.1/Admin** *Management of security attributes*

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/ Admin	The TSF shall enforce the SCD/SVD Generation SFP and the SCD Import SFP to restrict the ability to modify the security attributes SCD/SVD management to R.Admin.

8.1.2.1.20 **FMT_MSA.2** *Secure security attributes*

Hierarchical to:	No other components
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for: (1) SCD/SVD Management (2) SCD operational.

8.1.2.1.21 **FMT_MSA.3** *Static attribute initialisation*

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the SCD/SVD Generation SFP, SVD Transfer SFP, SCD import SFP, Signature Creation SFP, to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the authorized identified role to specify alternative initial values to override the default values when an object or information is created.

Application note:
The authorized identified roles are defined in the following table depending on the TOE lifecycle phase

Security attribute	Phase	Authorized identified roles
SCD/SVD Management	6&7	R.Admin
SCD Operational	7	R.Sigy

8.1.2.1.22 **FMT_MSA.4** *Security attribute value inheritance*

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes:
(1) If S.Admin successfully generates an SCD/SVD pair without S.Sigy being authenticated the security attribute "SCD operational" of the SCD shall be set to "no" as a single operation.
(2) If S.Admin imports SCD while S.Sigy is not currently authenticated, the security attribute "SCD operational" of the SCD shall be set to "no" after import of the SCD as a single operation
(3) If S.Admin imports SCD while S.Sigy is currently authenticated, the security attribute "SCD operational" of the SCD shall be set to "yes" after import of the SCD as a single operation

Application note:
(3) doesn't apply as SCD import is only possible in phase 6 where the role R.Sigy doesn't exist

8.1.2.1.23 **FMT_MTD.1/Admin** *Management of TSF data*

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/Admin The TSF shall restrict the ability to create the RAD to R.Admin.

8.1.2.1.24 **FPT_EMS.1** *TOE Emanation*

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_EMS.1.1 The TOE shall not emit side channel emission in excess of limits specified by the state of the art attacks on smart card IC enabling access to RAD and SCD.

FPT.EMS.1.2 The TSF shall ensure all users are unable to use the following interface external contacts emanations to gain access to RAD and SCD.

8.1.2.1.25 **FPT_FLS.1** *Failure with preservation of secure state*

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:
(1) self-test according to FPT_TST fails
(2) Security violation detected by [PLT] with FAU_ARP.1

8.1.2.1.26 **FPT_PHP.1** *Passive detection of physical attack*

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_PHP.1.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

8.1.2.1.27 **FPT_PHP.3** *Resistance to physical attack*

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_PHP.3.1	The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced.

8.1.2.1.28 **FPT_TST.1** *TSF testing*

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self-tests during initial start-up to demonstrate the correct operation of the TSF.
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of TSF data.
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of TSF.

8.1.2.1.29 **FTP_ITC.1/SCD** *Inter-TSF trusted channel*

Hierarchical to:	No other components.
Dependencies:	No Dependencies
FTP_ITC.1.1/SCD	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/SCD	The TSF shall permit another trusted IT product to initiate communication via the trusted channel
FTP_ITC.1.3/SCD	The TSF shall initiate communication via the trusted channel for (1) data exchange integrity according to FDP_UCT.1/SCD (2) none

8.1.2.1.30 **FTP_ITC.1/SVD** *Inter-TSF trusted channel*

Hierarchical to:	No other components.
Dependencies:	No Dependencies
FTP_ITC.1.1/SVD	The TSF shall provide a communication channel between itself and another trusted IT product **CGA** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/SVD	The TSF shall permit another trusted IT product to initiate communication via the trusted channel
FTP_ITC.1.3/SVD	The TSF **or the CGA** shall initiate communication via the trusted channel for (1) data Authentication with Integrity of Guarantor according to FIA_API.1 and FDP_DAU.2/SVD (2) none

8.1.2.2 Phase 7

8.1.2.2.1 **FCS_COP.1/Sign** *Cryptographic operation*

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/Sign — The TSF shall perform Signature Computation in accordance with a specified cryptographic algorithm [algorithm] and cryptographic key sizes [key size(s)] that meets the following [standard].

Algorithm	**key size(s)**	**standard**
Signature Computation with Off-Card Hashing: RSA and ECDSA	• RSA-1024, 1536 and 2048 with PKCS#1 v1.5 and PKCS#1-PSS • EC-DSA over elliptic curves of size of-192, 224, 256, 320, 384, 512 and 521 bits • SHA-1, 224, 256, 384 and 512	[PKCS#1] [ANSIX9.62]
Signature Computation with On-Card Hashing: ECDSA only	• EC-DSA over elliptic curves of size of 192, 224, 256, 320, 384, 512 and 521 • SHA-1, 256 and 384	

8.1.2.2.2 **FDP_ACC.1/Signature_Creation** *Subset access control*

Hierarchical to:	No other components
Dependencies:	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Signature_Creation — The TSF shall enforce the Signature Creation SFP on
(1) subjects: S.User,
(2) objects: DTBS/R, SCD,
(3) operations: signature creation.

8.1.2.2.3 **FDP_ACF.1/Signature creation** *Security attribute based access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/ Signature_Creation	The TSF shall enforce the Signature Creation SFP to objects based on the following: (1) the user S.User is associated with the security attribute "Role" and (2) the SCD with the security attribute "SCD Operational".
FDP_ACF.1.2/ Signature_Creation	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: R.Sigy is allowed to create electronic signatures for DTBS/R with SCD which security attribute "SCD operational" is set to "yes".
FDP_ACF.1.3/ Signature_Creation	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/ Signature_Creation	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: S.User is not allowed to create electronic signatures for DTBS/R with SCD which security attribute "SCD operational" is set to "no".

8.1.2.2.4 **FDP_SDI.2/DTBS** *Stored data integrity monitoring and action*

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.

FDP_SDI.2.1/DTBS	The TSF shall monitor user data stored in containers controlled by the TSF for integrity error on all objects, based on the following attributes: integrity checked stored DTBS.
FDP_SDI.2.2/DTBS	Upon detection of a data integrity error, the TSF shall (1) prohibit the use of the altered data (2) inform the S.Sigy about integrity error.

Application note:
The DTBS/R temporarily stored by the TOE has the user data attribute "integrity checked stored data".

8.1.2.2.5 **FIA_AFL.1/RAD** *Authentication failure handling*

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/RAD	The TSF shall detect when an administrator configurable positive integer within 1 and 15 unsuccessful authentication attempts occur related to consecutive failed authentication attempts.
FIA_AFL.1.2/RAD	When the defined number of unsuccessful authentication attempts has been met, the TSF shall block RAD.

Application note:
These SFRs apply to R.Sigy and R.Admin using the PUK to authenticate itself.

8.1.2.2.6 **FMT_MOF.1** *Management of security functions behavior*

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions.

FMT_MOF.1.1 The TSF shall restrict the ability to enable the functions signature creation function to R.Sigy.

8.1.2.2.7 **FMT_MSA.1/Signatory** *Management of security attributes*

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/Signatory The TSF shall enforce the Signature Creation SFP to restrict the ability to modify the security attributes SCD operational to R.Sigy.

8.1.2.2.8 **FMT_MTD.1/Signatory** *Management of TSF data*

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/Signatory The TSF shall restrict the ability to modify and none the RAD to R.Sigy.

Application note:
This requirement applies only to the RAD belonging to S.Sigy.

8.1.3 Additional SFRs

8.1.3.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/Session keys The TSF shall generate session keys in accordance with a specified cryptographic key generation algorithm: Key derivation function and specified cryptographic key sizes:

(1) DES keys of 112 bits
(2) AES keys of 128, 192 and 256 bits

that meet the following: [TR_03110], [GP2.3], [SCP03]

FCS_CKM.1.1/DH_PACE The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*cryptographic key generation algorithm]* and specified cryptographic key sizes *[cryptographic key sizes]* that meet the following: *[standard]*

cryptographic key generation algorithm	cryptographic key sizes	standard
ECDH compliant to [ISO_15946]	192 bits to 521 bits	Based on ECDH protocol compliant to [TR_03111]
DH compliant to	1024, 1536 and 2048 bits	Based on the Diffie-Hellman key derivation protocol compliant to [PKCS#3]

8.1.3.2 FCS_COP.1 *Cryptographic operation*

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/GP secret data protection The TSF shall perform GP secret data encryption in accordance with a specified cryptographic algorithm:
- SCP02
- SCP03 using AES

and cryptographic key sizes:
- 128 bits
- 128, 192, and 256 bits

that meet the following:
- [GP2.3]
- [SCP03]

Application Note 1:
The type of algorithm used by the TOE depends on the configuration set during the javacard open platform Personalization (For more details see [AGD_PRE_PLT]).

Application Note 2:
The applet provides this service via the platform, it doesn't own and cannot access the keys used to protect secret data. Their import/generation and destruction are managed by the platform.

FCS_COP.1.1/SM in confidentiality The TSF shall perform Secure messaging in confidentiality in accordance with a specified cryptographic algorithm:
(1) Encryption with TDES EDE in CBC mode
(2) Encryption with AES in CBC mode

and cryptographic key sizes:
(1) 128 bits
(2) 128 bits, 192 bits and 256 bits

that meet the following:
(1) [GP2.3] and [TR_03110]
(2) [SCP03] and [TR_03110]

Application Note:
This algorithm is used during secure Messaging to ensure confidentiality of incoming and outgoing data.

FCS_COP.1.1/SM in integrity

The TSF shall perform Secure messaging in integrity and authenticity in accordance with a specified cryptographic algorithm:
(1) Retail MAC: MAC algorithm 3 with padding method 2 and DES bloc Cipher
(2) CMAC
and cryptographic key sizes:
(1) 128 bits
(2) 128 bits, 192 bits and 256 bits
that meet the following:
(1) [GP2.3] and, [TR_03110]
(2) [SCP03] and [TR_03110]

Application Note:
This algorithm is used during secure Messaging to ensure integrity and authenticity of incoming and outgoing data.

FCS_COP.1.1/Digital Auth

The TSF shall perform Digital Authentication in accordance with a specified cryptographic algorithm [algorithm] and cryptographic key sizes [key size(s)] that meets the following [standard].

algorithm	key size(s)	standard
Digital authentication with Off-Card Hashing: RSA and ECDSA	• RSA-1024, 1536 and 2048 with PKCS#1 v1.5 and PKCS#1-PSS • EC-DSA over elliptic curves of size of-192, 224, 256, 320, 384, 512 and 521 bits • SHA-1, 224, 256, 384 and 512	[PKCS#1] [ANSIX9.62]
Digital authentication with On-Card Hashing: ECDSA only	• EC-DSA over elliptic curves of size of 192, 224, 256, 320, 384, 512 and 521 • SHA-1, 256 and 384	

FCS_COP.1.1/Enc key decipherment

The TSF shall perform Encryption key decipherment in accordance with a specified cryptographic algorithm [algorithm] and cryptographic key sizes [key size(s)] that meets the following [standard].

algorithm	key size(s)	standard
Encryption key decipherment: RSA	• RSA-1024, 1536 and 2048 with PKCS#1 OAEP, using SHA-256	[PKCS#1] [ANSIX9.62]
Encryption key decipherment: EC-DH	• EC-DH over elliptic curves of size of 192, 224, 256, 320, 384, 512 and 521, using SHA-1, 224, 256, 384 and 512	

FCS_COP.1.1/ SIG_VER

The TSF shall perform digital signature verification in accordance with a specified cryptographic algorithm [algorithm] and cryptographic key sizes [key size(s)] that meets the following [standard].

algorithm	key size(s)	standard
EC-DSA with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	• Elliptic curves of size of 192 to 521 bits	[TR_03110]
RSA PKCS#1 v1.5 & v2.1 PSS with SHA-1, SHA-256,SHA-512	• 1024 up to 2048 bits	

8.1.3.3 FCS_RND.1 Quality Metric for Random Numbers

Hierarchical to: No other components.
Dependencies: No dependencies

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet FCS_RNG.1 Quality metric for random numbers of [PLT].

8.1.3.4 FIA_UID.1/PACE

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1/PACE The TSF shall allow

1. to establish the communication channel
2. to carry out the PACE Protocol (PIN, PUK or CAN) according to [TR_03110] and/or the VERIFY PIN command
3. to read the Initialization Data if it is not disable by TSF
4. to carry out the Chip Authentication Protocol v.1 according to [TR_03110]
5. to carry out the Terminal Authentication Protocol v.1 according to [TR_03110]

on behalf of the user to be performed before the user is identified.

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

8.1.3.5 FIA_UAU.1/PACE Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/PACE The TSF shall allow

1. to establish the communication channel
2. to carry out the PACE Protocol (PIN, PUK or CAN) according to [TR_03110] and/or the VERIFY PIN command
3. to read the Initialization Data if it is not disable by TSF
4. to identify themselves by selection of the authentication key
5. to carry out the Chip Authentication Protocol v.1 according to [TR_03110]
6. to carry out the Terminal Authentication Protocol v.1 according to [TR_03110]

on behalf of the user to be performed before the user is authenticated.

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

8.1.3.6 FIA_UAU.4/ PACE Single-use authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.4.1/PACE The TSF shall prevent reuse of authentication data related to

1. PACE Protocol (PIN, PUK or CAN) according to [TR_03110]
2. Authentication Mechanisms based on Triple-DES or AES
3. Terminal Authentication Protocol v.1 according to [TR_03110]

Application Note:
The Authentication Mechanisms based on Triple-DES or AES is the authentication process performed in phases 5 and 6.

8.1.3.7 FIA_UAU.5/ PACE Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1/PACE The TSF shall provide

1. PACE Protocol (PIN, PUK or CAN) according to [TR_03110] and/or VERIFY PIN command
2. Mean to verify the integrity and authenticity of the Chip authentication public key
3. Secure messaging in MAC-ENC mode according to [TR_03111]
4. Symmetric Authentication Mechanism based on Triple-DES or AES
5. Terminal Authentication Protocol v.1 according to [TR_03110]

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

1. Having successfully run in contactless the PACE protocol (PIN, PUK or CAN) and/or the VERIFY PIN command, the TOE accepts only received commands with correct message authentication code sent by means of secure messaging with the key agreed with the terminal by means of the PACE protocol.
2.The establishment of the secure messaging with the PACE protocol is not mandatory if the VERIFY PIN command is used in contact mode.
3. The TOE accepts the authentication attempt as Personalization Agent by the Authentication Mechanism with Personalization Agent Key(s).
4 After run of the Chip Authentication Protocol Version 1, the TOE accepts only received commands with correct message authentication code sent by means of secure messaging with key agreed with the terminal by means of the Chip Authentication Mechanism v1.
5. The TOE accepts the authentication attempt by means of the Terminal Authentication Protocol v.1 only if the terminal uses the public key presented during the Chip Authentication Protocol v.1 and the secure messaging established by the Chip Authentication Mechanism v.1
6.The TOE accepts the authentication attempt as Personalization Agent by the Authentication Mechanism with Personalization Agent Key(s).

8.1.3.8 FIA_UAU.6/ PACE Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/PACE The TSF shall re-authenticate the user under the conditions each command sent to the TOE after successful run of the PACE protocol shall be verified as being sent by the BAT.

8.1.3.9 FIA_UAU.6/EAC Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/EAC The TSF shall re-authenticate the user under the conditions each command sent to the TOE after successful run of the Chip Authentication Protocol Version 1 shall be verified as being sent by the Inspection System.

8.1.3.10 FIA_AFL.1/AUTH Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1/AUTH The TSF shall detect when [selection] unsuccessful authentication attempts occur related to [list of authentication events].

FIA_AFL.1.2/AUTH When the defined number of unsuccessful authentication attempts has been met, the TSF shall [list of actions].

selection	list of authentication events	list of actions
Positive integer number set to 0x0A	Authentication attempt involving CAN as shared password for PACE	Wait for an increasing time between receiving of the terminal challenge and sending of the TSF response during the PACE authentication attempts
An administrator configurable positive integer within range of acceptable values 0 to 14 consecutive	Consecutive failed authentication attempts using the PIN or PUK as the shared password for PACE leaving a single authentication attempt	Suspend the PIN or the PUK in contactless
'1'	Consecutive failed authentication attempts using the suspended PIN or PUK as the shared password for PACE in contactless mode	Block the PIN or the PUK
An administrator configurable positive integer within range of acceptable values 0 to 15 consecutive	Consecutive failed authentication attempts using the PIN or PUK with VERIFY PIN command	Block the PIN or the PUK
'1'	Personalization agent authentication attempt	slow down exponentially the next authentication

8.1.3.11 FIA_API.1/TOE Authentication Proof of Identity

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1/TOE The TSF shall provide an authentication mechanism to prove the identity of the Document holder.

Application Note:
The TOE acts as a substitute for the Document holder, to authenticate digitally on its behalf. The authentication mechanism is triggered by the Document holder itself by presenting its PIN to the TOE.

8.1.3.12 FDP_ACC.1/TRM Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/TRM The TSF shall enforce the Access Control SFP on terminals gaining access User data stored in the TOE (including sensitive user data).

8.1.3.13 FDP_ACF.1/TRM Basic Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/TRM The TSF shall enforce the Access Control SFP to objects based on the following:
- Subjects: Terminal, BAT, Authentication Terminal.
- Objects: User data stored in the TOE (including sensitive user data),
- Security attributes: Terminal Authorization.

FDP_ACF.1.2/TRM The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
- A BAT is allowed to read data objects (except sensitive user data) specified in FDP_ACF.1.1/TRM after successful authentication, as required by FIA_UAU.1/PACE.
- Reading, modifying, writing, or using sensitive user data protected by CAv1 and TAv1 (objects specified in FDP_ACF.1.1/TRM) is only allowed to Authentication Terminals using the following mechanism: The TOE applies the EAC protocol (cf. FIA_UAU.5) to determine effective authorizations of the terminal. To determine the effective authorization of a terminal, the chip must calculate a bitwise Boolean 'and' of the relative authorization contained in the CHAT of the Terminal Certificate, the referenced eService certification authority Certificate. Based on the terminal type drawn from the CHAT of the Terminal Certificate, the TOE shall grant the right to read, modify or write sensitive user data, or perform operations using these sensitive user data.

FDP_ACF.1.3/TRM The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.

FDP_ACF.1.4/TRM The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
- Any terminal not being a BAT or an Authentication Terminal is not allowed to read, to write, to modify, or to use any user data specified in FDP_ACF.1.1/TRM.
- In contactless, terminals not using secure messaging are not allowed to read, write, modify, or use any user data specified in FDP_ACF.1.1/TRM.

8.1.3.14 FDP_UCT.1/TRM Basic data exchange confidentiality

Hierarchical to: No other components.

Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
[FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_UCT.1.1/TRM The TSF shall enforce the Access Control SFP to be able to transmit and receive user data in a manner protected from unauthorized disclosure.

8.1.3.15 FDP_UIT.1/TRM Data exchange integrity

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]

FDP_UIT.1.1/TRM The TSF shall enforce the Access Control SFP to be able to transmit and receive user data in a manner protected from modification, deletion, insertion and replay errors.

FDP_UIT.1.2/TRM The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion and replay has occurred.

8.1.3.16 FTP_ITC.1/PACE Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No Dependencies.

FTP_ITC.1.1/PACE **[Editorially Refined]** The TSF shall provide a communication channel between itself and a BAT (in contactless mode) that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. The trusted channel shall be established by performing the PACE protocol according to [TR_03110].

FTP_ITC.1.2/PACE **[Editorially Refined]**	The TSF shall permit the BAT (in contactless mode) to initiate communication via the trusted channel.
FTP_ITC.1.3/PACE	The TSF shall initiate communication via the trusted channel for any data exchange between the TOE and a BAT after PACE (in contactless mode).

8.1.3.17 FMT_SMR.1/ PACE Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FMT_SMR.1.1/PACE The TSF shall maintain the roles:

- Personalization Agent,
- Terminal,
- BAT,
- Country Verifying Certification Authority,
- eService certification authority,
- Authentication Terminal,
- Document holder.

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

8.1.3.18 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MTD.1.1/CVCA_INI	The TSF shall restrict the ability to write the 1. initial Country Verifying Certification Authority Public Key, 2. initial Country Verifying Certification Authority Certificate, 3. initial Current Date to the Personalization Agent.
FMT_MTD.1.1/CVCA_UPD	The TSF shall restrict the ability to update the 1. Country Verifying Certification Authority Public Key, 2. Country Verifying Certification Authority Certificate, to Country Verifying Certification Authority.
FMT_MTD.1.1/DATE	The TSF shall restrict the ability to modify the Current date to 1. Country Verifying Certification Authority, 2. eService certification authority, 3. Authentication Terminal.
FMT_MTD.1.1/CAPK	The TSF shall restrict the ability to load or generate the Chip Authentication Private Key selected in Access Control SFP to the Personalization Agent.

FMT_MTD.1.1/KEY_READ	The TSF shall restrict the ability to read the 1. PACE passwords, 2. Personalization Agent Keys, 3. Chip Authentication Private Key, to none
FMT_MTD.1.1/Initialize_PIN	The TSF shall restrict the ability to write the PIN, PUK and CAN, selected in Access Control SFP to the personalization agent.
FMT_MTD.1.1/Resume_PIN	The TSF shall restrict the ability to resume[2] the suspended PIN or PUK selected in Access Control SFP (in contactless mode) to the electronic document presenter.
FMT_MTD.1.1/Change_PIN	The TSF shall restrict the ability to change the PIN selected in Access Control SFP to the document holder (using the Current PIN value for changing or the PUK).
FMT_MTD.1.1/Unblock_PIN	The TSF shall restrict the ability to unblock the PIN selected in Access Control SFP to the document holder (using the PUK for unblocking).
FMT_MTD.1.1/UnblockChange_RAD	The TSF shall restrict the ability to unblock and optionally change the RAD selected in Access Control SFP to o If change is required, the document holder (using the PUK for unblocking) and an Authentication Terminal (TA_PMT) in accordance with FMT_MTD.1/Signatory; o Otherwise if only unblock is required, the document holder (using the PUK for unblocking).
FMT_MTD.1.1/Erase_PIN	The TSF shall restrict the ability to erase the PIN or RAD selected in Access Control SFP to an Authentication Terminal (TA_PMT).
FMT_MTD.1.1/Reinitialize_PIN	The TSF shall restrict the ability to (re)initialize the PIN selected in Access Control SFP to the document holder (using the PUK).
FMT_MTD.1.1/UnblockChange_PUK	The TSF shall restrict the ability to unblock and change the PUK selected in Access Control SFP to an Authentication Terminal (TA_PMT).
FMT_MTD.1.1/TOE state	The TSF shall restrict the ability to switch the TOE from phase 6 to phase 7 to the personalization agent.

[2] "Resume" is called also "De-suspend" in guidance documents.

8.1.3.19 FMT_MTD.3 Secure TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data

FMT_MTD.3.1 The TSF shall ensure that only secure values of the certificate chain are accepted for TSF data of the Terminal Authentication Protocol v1 and the Access Control SFP.

8.1.3.20 FMT_LIM.1 Limited capabilities

Hierarchical to: No other components.

Dependencies: FMT_LIM.2 Limited availability.

FMT_LIM.1.1 The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced**:**

Deploying Test Features after TOE Delivery does not allow
1. User Data to be disclosed and manipulated,
2. TSF data to be disclosed or manipulated,
3. software to be reconstructed and,
4. substantial information about construction of TSF to be gathered which may enable other attacks.

8.1.3.21 FMT_LIM.2 Limited availability

Hierarchical to: No other components.

Dependencies: FMT_LIM.1 Limited capabilities.

FMT_LIM.2.1 The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced**:**

Deploying Test Features after TOE Delivery does not allow
1. User Data to be disclosed and manipulated,
2. TSF data to be disclosed or manipulated,
3. software to be reconstructed and,
4. substantial information about construction of TSF to be gathered which may enable other attacks.

8.1.3.22 FPT_EMS.1/PIN-PUK-KEYS TOE Emanation

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_EMS.1.1/ PIN-PUK-KEYS The TOE shall not emit side channel emission in excess of limits specified by the state of the art attacks on smart card IC enabling access to PIN, PUK, Keys and none.

FPT.EMS.1.2/ PIN-PUK-KEYS The TSF shall ensure all users are unable to use the following interface external contacts emanations to gain access to PIN, PUK, Keys and none.

8.2 Security Assurance Requirements

Assurance class	Assurance components
ADV: Development	ADV_ARC.1: Security architecture description ADV_FSP.5: Complete semi-formal functional specification with additional error information ADV_IMP.1: Implementation representation of the TSF ADV_INT.2: well-structured internals ADV_TDS.4: Semiformal modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance AGD_PRE.1: Preparative procedures
ALC: Life Cycle Support	ALC_CMC.4: Production support, acceptance procedures and automation ALC_CMS.5: Development tools CM coverage ALC_DEL.1: Delivery procedures ALC_DVS.2: Identification of security measures (augmented) ALC_LCD.1: Developer defined life cycle model ALC_TAT.2: Compliance with implementation standards
ASE: Security Target Evaluation	ASE_CCL.1: Conformance Claims ASE_ECD.1: Extended components definition ASE_INT.1: ST introduction ASE.OBJ.2: Security Objectives ASE.REQ.2: Derived security requirements ASE.SPD.1: Security problem definition ASE_TSS.1: TOE summary specification
ATE: Tests	ATE_COV.2: Analysis of Coverage ATE_DPT.3: Testing modular design ATE_FUN.1: Functional Testing ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.5: Methodical vulnerability analysis (augmented)

Table 6- EAL5 +

8.2.1 AVA_VAN.5 augmentation

The TOE is intended to function in a variety of signature creation systems for sualified electronic signatures. Due to the nature of its intended applications, i.e. the TOE may be issued to users and may not be directly under the control of trained and dedicated administrators. Insecure states shall be easy to detect and the TOE shall be shown to be highly resistant to penetration attacks to meet the security objectives OT.SCD_Secrecy, OT.Sigy_SigF, OT.Sig_Secure and OT.Keys_Secrecy.

This assurance requirement is achieved by the AVA_VAN.5 component. Independent vulnerability analysis is based on highly detailed technical information. The attacker is assumed to be thoroughly familiar with the specific implementation of the TOE. The attacker is presumed to have a high level of technical sophistication.

8.2.2 ALC_DVS.2 augmentation

In order to protect the TOE on development Phase, the component ALC_DVS.2 was added. This latter requires security documentation justifying that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

8.3 Security Requirements Rationale

8.3.1 Security requirement coverage

Functional Requirements \ TOE security objectives	OT.lifecycle_Security	OT.SCD/SVD_Auth_Gen	OT.SCD_Unique	OT.SCD_SVD_Corresp	OT.SCD_Auth_Imp	OT.SCD_Secrecy	OT.Sig_Secure	OT.Sigy_SigF	OT.DTBS_Integrity_TOE	OT.EMSEC_Design	OT.Tamper_ID	OT.Tamper_Resistance	OT.TOE_SSCD_Auth	OT.TOE_TC_SVD_Exp	OT.Authentication_Secure	OT.Key_Lifecycle_Security	OT.Keys_Secrecy	OT.TOE_AuthKey_Uniq	OT.Lifecycle_Management	OT.eServices	OT.AC_Pers_EAC	OT.Tracing
FCS_CKM.1/SCD/SVD_Generation	X		X	X		X										X	X	X				
FCS_CKM.1/Session_keys															X							
FCS_CKM.4	X					X									X	X	X					
FCS_COP.1/Sign	X						X															
FCS_COP.1/GP secret data protection								X							X				X			
FCS_COP.1/SM in confidentiality															X							
FCS_COP.1/SM in integrity															X							
FCS_COP.1/Digital Auth																X				X		
FCS_COP.1/Enc key decipherment																X				X		
FCS_RND.1								X							X				X			
FDP_ACC.1/SCD/SVD_Generation	X	X																	X			
FDP_ACC.1/SCD_import	X				X														X			
FDP_ACC.1/SVD_Transfer	X													X					X			
FDP_ACC.1/Signature_creation	X							X											X			
FDP_ACF.1/SCD/SVD_Generation	X	X																	X			
FDP_ACF.1/SVD_Transfer	X													X					X			
FDP_ACF.1/SCD_import	X				X														X			
FDP_ACF.1/Signature_creation	X							X											X			
FDP_RIP.1						X		X							X		X				X	
FDP_SDI.2/Persistent				X		X	X								X		X					
FDP_SDI.2/DTBS								X	X													
FDP_ITC.1/SCD	X																					
FDP_UCT.1/SCD	X					X																
FDP_DAU.2/SVD														X								
FIA_AFL.1/RAD								X							X					X		
FIA_UAU.1		X			X			X					X		X				X	X		
FIA_UID.1		X			X			X							X				X	X		
FIA_API.1													X		X							
FMT_MOF.1	X							X											X			
FMT_MSA.1/Admin	X	X																	X			
FMT_MSA.1/Signatory	X							X														
FMT_MSA.2	X	X						X								X			X			
FMT_MSA.3	X	X						X								X			X			

FMT_MSA.4	X	X		X				X														
FMT_MTD.1/Admin	X							X											X			
FMT_MTD.1/Signatory	X							X											X			
FMT_MTD.1/TOE state																			X			
FPT_EMS.1						X				X					X							
FMT_SMR.1	X							X							X	X			X	X		
FMT_SMF.1	X			X				X							X	X			X	X	X	
FPT_FLS.1						X									X		X					
FPT_PHP.1											X											
FPT_PHP.3						X						X			X		X					
FPT_TST.1	X					X	X								X	X	X					
FTP_ITC.1/SCD	X					X																
FTP_ITC.1/SVD														X								
FCS_CKM.1.1/DH_PACE															X							
FCS_COP.1.1/SIG_VER															X							
FIA_UAU.1/PACE															X						X	
FIA_UAU.5/PACE															X							
FIA_AFL.1/AUTH															X						X	X
FIA_API.1/TOE															X							
FIA_UAU.6/CA															X							
FIA_UID.1/PACE															X						X	
FIA_UAU.4/PACE															X							
FIA_UAU.6/PACE															X							
FDP_ACF.1/TRM															X						X	
FDP_ACC.1/TRM															X						X	
FDP_UCT.1/TRM															X							
FDP_UIT.1/TRM															X							
FTP_ITC.1/PACE															X							X
FMT_SMR.1/PACE															X						X	
FMT_MTD.1.1/CVCA_INI															X							
FMT_MTD.1.1/CVCA_UPD															X							
FMT_MTD.1.1/DATE															X							
FMT_MTD.1.1/CAPK															X							
FMT_MTD.1.1/KEY_READ															X						X	
FMT_MTD.1.1/Initialize_PIN															X						X	
FMT_MTD.1.1/Resume_PIN															X						X	
FMT_MTD.1/Change_PIN															X						X	
FMT_MTD.1/Unblock_PIN								X							X				X		X	
FMT_MTD.1.1/UnblockChange_RAD	X							X							X				X		X	
FMT_MTD.1.1/Erase_PIN	X														X				X		X	
FMT_MTD.1.1/Reinitialize_PIN															X						X	
FMT_MTD.1.1/UnblockChange_PUK															X						X	
FMT_MTD.3															X							
FMT_LIM.1															X							
FMT_LIM.2															X							
FPT_EMS.1/PIN-PUK-KEYS										X					X		X					

8.3.2 TOE security requirements sufficiency

OT.Lifecycle_Security (*Lifecycle security*) is provided by the SFR as follows.
The SCD import is controlled by TSF according to **FDP_ACC.1/SCD_Import**, **FDP_ACF.1/SCD_Import** and **FDP_ITC.1/SCD**. The confidentiality of the SCD is protected during import according to **FDP_UCT.1/SCD** in the trusted channel **FTP_ITC.1/SCD**.

Secure SCD/SVD generation is ensured by **FCS_CKM.1/SCD/SVD_Generation**. The SCD/SVD generation is controlled by TSF according to **FDP_ACC.1/SCD/SVD_Generation** and **FDP_ACF.1/SCD/SVD_Generation**. The SVD transfer for certificate generation is controlled by TSF according to **FDP_ACC.1/SVD_Transfer** and **FDP_ACF.1/SVD_Transfer**.

The secure SCD usage is ensured cryptographically according to **FCS_COP.1/Sign.**The SCD usage is controlled by access control **FDP_ACC.1/Signature_Creation**, **FDP_ACF.1/Signature_Creation** which is based on the security attribute secure TSF management according to **FMT_MOF.1**, **FMT_MSA.1/Admin**, **FMT_MSA.1/Signatory, FMT_MSA.2, FMT_MSA.3, FMT_MSA.4**, **FMT_MTD.1/Admin**, **FMT_MTD.1/Signatory**, ***FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN,*** **FMT_SMF.1** and **FMT_SMR.1**. The test functions **FPT_TST.1** provides failure detection throughout the lifecycle.

The SFR **FCS_CKM.4,** ensures a secure SCD desctruction.

OT.SCD/SVD_Auth_Gen (*Authorized SCD/SVD generation*) addresses that generation of a SCD/SVD pair requires proper user authentication. The TSF specified by **FIA_UID.1** and **FIA_UAU.1** provide user identification and user authentication prior to enabling access to authorized functions. The SFR **FDP_ACC.1/SCD/SVD_Generation** and **FDP_ACF.1/SCD/SVD_Generation** provide access control for the SCD/SVD generation. The security attributes of the authenticated user are provided by **FMT_MSA.1/Admin**, **FMT_MSA.2**, and **FMT_MSA.3** for static attribute initialisation. The SFR **FMT_MSA.4** defines rules for inheritance of the security attribute “SCD operational” of the SCD.

OT.SCD_Unique (*Uniqueness of the signature creation data*) implements the requirement of practically unique SCD as laid down in Annex III, paragraph 1(a), which is provided by the cryptographic algorithms specified by **FCS_CKM.1/SCD/SVD_Generation**

OT.SCD_SVD_Corresp *(Correspondence between SVD and SCD)* addresses that the SVD corresponds to the SCD implemented by the TOE. This is provided by the algorithms specified by **FCS_CKM.1/SCD/SVD_Generation** to generate corresponding SVD/SCD pairs. The security functions specified by **FDP_SDI.2/Persistent** ensure that the keys are not modified, so to retain the correspondence. Moreover, the SCD Identifier allows the environment to identify the SCD and to link it with the appropriate SVD. The management functions identified by **FMT_SMF.1** and by **FMT_MSA.4** allow R.Admin to modify the default value of the security attribute SCD Identifier.

OT.SCD_Auth_Imp (*Authorized SCD import*) is provided by the security functions specified by the following SFR. **FIA_UID.1** and **FIA.UAU.1** ensure that the user is identified and authenticated before SCD can be imported. **FDP_ACC.1/SCD_Import** and **FDP_ACF.1/SCD_Import** ensure that only authorised users can import SCD.

OT.SCD_Secrecy (*Secrecy of signature creation data)* is provided by the security functions specified by the following SFR. **FDP_UCT.1/SCD** and **FTP_ITC.1/SCD** ensures the confidentiality for SCD import.

FCS_CKM.1/SCD/SVD_Generation ensures the use of secure cryptographic algorithms for SCD/SVD generation. Cryptographic quality of SCD/SVD pair shall prevent disclosure of SCD by cryptographic attacks using the publicly known SVD.

The security functions specified **by FDP_RIP.1** and **FCS_CKM.4** ensure that residual information on SCD is destroyed after the SCD has been use for signature creation and that destruction of SCD leaves no residual information.

The security functions specified by **FDP_SDI.2/Persistent** ensure that no critical data is modified which could alter the efficiency of the security functions or leak information of the SCD. **FPT_TST.1** tests the working conditions of the TOE and **FPT_FLS.1** guarantees a secure state when integrity is violated and thus assures that the specified security functions are operational. An example where compromising error conditions are countered by **FPT_FLS.1** is fault injection for differential fault analysis (DFA).

SFR **FPT_EMS.1** and **FPT_PHP.3** require additional security features of the TOE to ensure the confidentiality of the SCD.

OT.Sig_Secure (*Cryptographic security of the electronic signature*) is provided by the cryptographic algorithms specified by **FCS_COP.1/Sign**, which ensures the cryptographic robustness of the signature algorithms. **FDP_SDI.2/Persistent** corresponds to the integrity of the SCD implemented by the TOE and **FPT_TST.1** ensures self-tests ensuring correct signature creation.

OT.Sigy_SigF (*Signature creation function for the legitimate signatory only*) is provided by an SFR for identification authentication and access control. **FIA_UAU.1** and **FIA_UID.1** ensure that no signature creation function can be invoked before the signatory is identified and authenticated.

The security functions specified by **FMT_MTD.1/Admin** and **FMT_MTD.1/Signatory** manage the authentication function. SFR **FIA_AFL.1/RAD** provides protection against a number of attacks, such as cryptographic extraction of residual information, or brute force attacks against authentication.

The security function specified by **FDP_SDI.2/DTBS** ensures the integrity of stored DTBS. The security functions specified by **FDP_ACC.1/Signature_Creation** and **FDP_ACF.1/Signature_Creation** provide access control based on the security attributes managed according to the SFR **FMT_MTD.1/Signatory**, **FMT_MSA.2, FMT_MSA.3** and **FMT_MSA.4**. The SFR **FMT_SMF.1** and **FMT_SMR.1** list these management functions and the roles. These ensure that the signature process is restricted to the signatory. **FMT_MOF.1** restricts the ability to enable the signature creation function to the signatory.

FMT_MSA.1/Signatory restricts the ability to modify the security attributes SCD operational to the signatory. Furthermore, **FDP_RIP.1** prevents misuse of any resources containing the SCD after de-allocation (e.g. after the signature creation process) and ensures that no attacker can get hold of the SCD (to create signatures outside the TOE) once SCD has been deleted by the legitimate signatory.

FMT_MTD.1/Unblock_PIN *and* **FMT_MTD.1.1/UnblockChange_RAD** *ensure the unblocking of the PIN (including the RAD) is made under the sole control of the administrator.*
In phase 6, the RAD may be loaded on the TOE by the Personalization Agent as defined in ***FMT_SMF.1****. The Personalization Agent is authenticated with a mutual authentication performed with* ***FCS_RND.1*** *and* ***FCS_COP.1/GP,*** *and is authenticated with* ***FMT_SMR.1****. During the mutual authentication, a session encryption key is agreed between the TOE and the Personalization Agent and used by the TOE to decrypt the RAD using* ***FCS_COP.1/GP secret data Protection****, ensuring the confidentiality of the RAD during its transfer in phase 6.*
In phase 6, ***FMT_MSA.1/ Signatory*** *guarantees that the Personalization Agent cannot sign on behalf of the signatory, ensuring the signature creation features remains under the sole control of the signatory.*

OT.DTBS_Integrity_TOE (*DTBS/R integrity inside the TOE)* ensures that the DTBS/R is not altered by the TOE. The integrity functions specified by **FDP_SDI.2/DTBS** require that the DTBS/R has not been altered by the TOE.

OT.EMSEC_Design (*Provide physical emanations security*) covers that no intelligible information is emanated. This is provided by **FPT_EMS.1.1** ***and FPT_EMS.1/PIN-PUK-KEYS.***

OT.Tamper_ID (*Tamper detection*) is provided by **FPT_PHP.1** by the means of passive detection of physical attacks.

OT.Tamper_Resistance (*Tamper resistance*) is provided by **FPT_PHP.3** to resist physical attacks.

OT.TOE_SSCD_Auth (*Authentication proof as SSCD*) requires the TOE to provide security mechanisms to identify and to authenticate themselves as SSCD, which is directly provided by **FIA_API.1** (*Authentication proof of identity*). The SFR **FIA_UAU.1** allows establishment of the trusted channel before (human) user is authenticated.

OT.TOE_TC_SVD_Exp (*TOE trusted channel for SVD export*) requires the TOE to provide a trusted channel to the CGA to protect the integrity of the SVD exported to the CGA, which is directly provided by

- The SVD transfer for certificate generation is controlled by TSF according to **FDP_ACC.1/SVD_Transfer** and **FDP_ACF.1/SVD_Transfer**
- **FDP_DAU.2/SVD** (*Data authentication with identity of guarantor*), which requires the TOE to provide CGA with the ability to verify evidence of the validity of the SVD and the identity of the user that generated the evidence.
- **FTP_ITC.1/SVD** (*inter-TSF trusted channel*), which requires the TOE to provide a trusted channel to the CGA.

OT.Authentication_Secure (*Secure authentication mechanisms*) is provided by **FCS_RND.1** and **FCS_COP.1/GP** for the authentication of the Personalization agent.

The use of a challenge freshly generated by the TOE with **FCS_RND.1** in theses authentication protocols ensures a protection against replay attacks when authenticating external entities. The security function specified by **FPT_TST.1** ensures that the security functions are performed correctly and **FDP_SDI.2/Persistent** guarantees the integrity of the authentication key(s) used by the TOE. **FMT_SMR.1** and **FMT_SMF.1** ensure the TOE can distinguish between external entities successfully authenticated (R.Admin) and can grant them dedicated rights.
In case of authentication protocols involving the import of ephemeral public key on the TOE (using Card verifiable certificates), **FDP_RIP.1** ensures that the key value is not kept by the TOE after usage and then can not be reused for a replay attack.

This objective ensures as well the establishment of a trusted channel following a successful mutual authentication. This trusted channel ensures authenticity, integrity and confidentiality of communication. **FCS_CKM.1/Session keys** generate session keys for the secure communication from a common secret agreed between the TOE and the external entity during the mutual authentication procedure.

Any incoming command shall contain a MAC computed by the issuer with the session key agreed during the mutual authentication, so that any unauthenticated or non integer command is detected by the MAC verification performed by the TOE using **FCS_COP.1/SM in integrity.** The data exchanged through this trusted channel are also protected in confidentiality thanks to **FCS_COP.1/SM in confidentiality**, ensuring they can only be disclosed to the parties authenticated during the mutual authentication step. The encryption key is ephemeral as it is generated during the mutual authentication using a challenge freshly generated by the TOE using **FCS_RND.1**, which ensures that dictionary attacks cannot be performed on encrypted data. When an integrity error is detected, or if the MAC is wrong (wrong authentication of the command issuer), the session keys (for integrity and confidentiality) are erased thanks to **FCS_CKM.4** so that they cannot be reused anymore, causing the trusted channel to be irreversibly lost. In particular, it ensures that encrypted data that may be caught by an attacker cannot be reused anymore to masquerade the TOE.

In phase 6, the integrity and confidentiality of data is ensured by **FCS_COP.1/GP secret data protection**.

The SSCD provides a proof of identity with **FIA_API.1**.

This objective ensures as well that any authentication key is loaded in the TOE by an authenticated user, so that only genuine keys associated to genuine users are declared to the TOE. The key import defined by **FMT_SMF.1** is protected by access control **.** It is protected by the proper Agent(s) authentication that cannot be bypassed to access these functions with the TSF specified by **FIA_UID.1** and **FIA_UAU.1.** The agent entitled to load the authentication key is (are) authenticated with **FMT_SMR.1.** Effort to bypass the access control by a frontal exhaustive attack is blocked by **FIA_AFL.1/RAD**.

This objective ensures the verification of the authenticity of the TOE as a whole device. This objective is mainly achieved by **FIA_API.1/TOE** using **FCS_CKM.1.1/DH_PACE**. CA provides an evidence of possessing the Chip Authentication Private Key (CAPK). **FMT_MTD.1.1/CAPK** governs creating/loading CAPK, whereas **FMT_MTD.1.1/KEY_READ** requires making this key unreadable by users. Hence, its value remains confidential. **FDP_RIP.1** requires erasing the values of CAPK and the session keys, here for CMAC. The authentication token is calculated using **FCS_COP.1.1/SM in integrity**.

The TOE holder provides a proof of identity with **FIA_API.1/TOE**.

This objective aims to explicitly protect sensitive (as opposed to common) user and TSF-Data. This is mainly achieved by enforcing (**FDP_UCT.1/TRM** and **FDP_UIT.1/TRM**) the access control SFPs **FDP_ACC.1/TRM** and **FDP_ACF.1/TRM**. A specific authorization level is achieved by terminal identification/authentication as required by the SFRs **FIA_UID.1/PACE**, **FIA_UAU.1/PACE** supported by **FCS_COP.1/SIG_VER**. The TA protocol uses the result of the PACE authentication (**FIA_UID.1/PACE**, **FIA_UAU.1/PACE**, confidentiality of the PACE passwords is ensured by **FMT_MTD.1.1/KEY_READ**) being, in turn, supported by **FCS_CKM.1.1/DH_PACE**. Since PACE can use the PIN as the shared secret, the use and management of the PIN (**FIA_AFL.1/AUTH, FMT_MTD.1.1/Resume_PIN, FMT_MTD.1/Unblock_PIN, FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1/Change_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK**) also support to achieve this objective. **FDP_RIP.1** requires erasing the temporal values of the PIN and PUK. **FIA_UAU.4/PACE, FIA_UAU.5/PACE, FIA_UAU.6/PACE** and **FCS_CKM.4** represent some specific properties of the used protocols. To allow for a verification of the certificate chain as required in

FMT_MTD.3, the CVCA's public key and certificate as well as the current date are written or updated by authorized identified role as required by **FMT_MTD.1.1/CVCA_INI, FMT_MTD.1.1/CVCA_UPD** and **FMT_MTD.1.1/DATE**. In contactless, this objective for the data exchanged is mainly achieved by **FTP_ITC.1/PACE** using **FCS_COP.1.1/SM in confidentiality**. A prerequisite for establishing this trusted channel is a successful Chip Authentication, cf. **FIA_API.1/TOE** using **FCS_CKM.1.1/ DH_PACE** and possessing the special properties **FIA_UAU.5/PACE**, and **FIA_UAU.6/CA**. As a prerequisite of this trusted channel, a trusted channel is established with the PACE protocol using **FIA_UID.1/PACE, FIA_UAU.1/PACE and FCS_CKM.1.1/DH_PACE** and possessing the special properties **FIA_UAU.5/PACE, FIA_UAU.6/PACE**. CA provides an evidence of possessing the Chip Authentication Private Key (CAPK). **FMT_MTD.1.1/CAPK** governs creating/loading CAPK, **FMT_MTD.1.1/KEY_READ** requires making this key unreadable by users. Thus its value remains confidential. FDP_RIP.1 requires erasing the values of CAPK and session keys, here for KENC. The SFRs **FMT_SMF.1** and **FMT_SMR.1/PACE** support the related functions and roles.

This objective ensures also the authenticity of user- and TSF-Data (after Terminal- and the Chip Authentication) by enabling its verification on both the terminal-side and by an active verification by the TOE itself. This objective is mainly achieved by FTP_ITC.1/PACE in contactless using FCS_COP.1.1/SM in integrity. A prerequisite for establishing this trusted channel is a successful Chip Authentication, cf. FIA_API.1/TOE using FCS_CKM.1.1/DH_PACE and possessing the special properties FIA_UAU.5/PACE, and FIA_UAU.6/CA. As a prerequisite of this trusted channel, a trusted channel is established with the PACE protocol using FIA_UID.1/PACE, FIA_UAU.1/PACE and FCS_CKM.1.1/DH_PACE and possessing the special properties FIA_UAU.5/PACE, FIA_UAU.6/PACE. CA provides an evidence of possessing the Chip Authentication Private Key (CAPK). FMT_MTD.1.1/CAPK governs creating/loading CAPK, FMT_MTD.1.1/KEY_READ requires to make this key unreadable by users. Hence its value remains confidential. FDP_RIP.1 requires to erase the values of CAPK and session keys, here for KMAC. A prerequisite for successful CA is an accomplished TA as required by FIA_UID.1/PACE, FIA_UAU.1/PACE supported by FCS_COP.1/SIG_VER. The TA protocol uses the result of the PACE authentication (FIA_UID.1/PACE, FIA_UAU.1/PACE) being, in turn, supported by FCS_CKM.1.1/DH_PACE. Since PACE can use the PIN as the shared secret, the use and management of the PIN (FIA_AFL.1/AUTH, FMT_MTD.1.1/Resume_PIN, FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1/Change_PIN, FMT_MTD.1/Unblock_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK) also support achieving this objective. FDP_RIP.1 requires to erase the temporal values of the PIN and PUK.FIA_UAU.4/PACE, FIA_UAU.5/PACE, FIA_UAU.6/CA and FCS_CKM.4 represent some specific required properties of the used protocols.To allow for a verification of the certificate chain as required in FMT_MTD.3, the CVCA's public key and certificate, as well as the current date, are written or updated by authorized identified roles as required by FMT_MTD.1.1/CVCA_INI, FMT_MTD.1.1/CVCA_UPD and FMT_MTD.1.1/DATE. The SFRs FMT_SMF.1 and FMT_SMR.1/PACE support the related functions and roles.
This objective ensures the confidentiality of the user- and TSF-Data stored and, after Terminal- and Chip Authentication, of their exchange. This objective for the data stored is mainly achieved by FDP_ACC.1/TRM and FDP_ACF.1/TRM. Enforcement of the two previous in a protected manner is ensured by FDP_UCT.1/TRM and FDP_UIT.1/TRM. A specific authorization level is achieved by terminal identification/authentication as required by the SFRs FIA_UID.1/PACE, FIA_UAU.1/PACE supported by FCS_COP.1/SIG_VER. The TA protocol uses the result of the PACE authentication (FIA_UID.1/PACE, FIA_UAU.1/PACE, confidentiality of the PACE passwords is ensured by FMT_MTD.1.1/KEY_READ) being, in turn, supported by FCS_CKM.1.1/DH_PACE. Since PACE can use the PIN as the shared secret, the use and management of the PIN (FIA_AFL.1/AUTH, FMT_MTD.1.1/Resume_PIN, FMT_MTD.1/Unblock_PIN, FMT_MTD.1/Change_PIN, FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK) also support to achieve this objective. FDP_RIP.1 requires erasing the temporal values of the PIN and PUK.FIA_UAU.4/PACE, FIA_UAU.5/PACE, FIA_UAU.6/PACE and FCS_CKM.4 represent some specific properties of the used protocols.To allow for a verification of the certificate chain as required in FMT_MTD.3, the CVCA's public key and certificate as well as the current date are written or updated by authorized identified role as required by FMT_MTD.1.1/CVCA_INI, FMT_MTD.1.1/CVCA_UPD and FMT_MTD.1.1/DATE. This objective for the data exchanged is mainly achieved in contactless by FTP_ITC.1/PACE using FCS_COP.1.1/SM in confidentiality. A prerequisite for establishing this trusted channel is a successful Chip Authentication, cf. FIA_API.1/TOE using FCS_CKM.1.1/DH_PACE and possessing the special properties FIA_UAU.5/PACE, and FIA_UAU.6/CA. As a prerequisite of this trusted channel, a trusted channel is established with the PACE protocol using FIA_UID.1/PACE, FIA_UAU.1/PACE and FCS_CKM.1.1/DH_PACE and possessing the special properties FIA_UAU.5/PACE, FIA_UAU.6/PACE. CA provides an evidence of possessing the Chip Authentication Private Key (CAPK). FMT_MTD.1.1/CAPK governs creating/loading CAPK, FMT_MTD.1.1/KEY_READ requires making this key unreadable by users. Thus its value remains confidential. FDP_RIP.1 requires erasing the values of CAPK and session keys, here for KENC. The SFRs FMT_SMF.1 and FMT_SMR.1/PACE support the related functions and roles.
This objective ensures the integrity of stored user- and TSF-Data and, after Terminal- and Chip Authentication, of these data exchanged (physical manipulation and unauthorized modifying). Physical manipulation is addressed by

FPT_PHP.3.Unauthorized modifying of the stored data is addressed by FDP_ACC.1/TRM and FDP_ACF.1/TRM. Enforcement of the two previous in a protected manner is ensured by FDP_UCT.1/TRM and FDP_UIT.1/TRM. A specific authorization level is achieved by terminal identification/ authentication as required by the SFRs FIA_UID.1/PACE, FIA_UAU.1/PACE supported by FCS_COP.1/SIG_VER. The TA protocol uses the result of PACE authentication (FIA_UID.1/PACE, FIA_UAU.1/PACE) being, in turn, supported by FCS_CKM.1.1/DH_PACE. Since PACE can use the PIN as the shared secret, using and management of PIN (FIA_AFL.1/AUTH, FMT_MTD.1.1/Resume_PIN, FMT_MTD.1/Change_PIN, FMT_MTD.1/Unblock_PIN FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK) also support achievement of this objective. FDP_RIP.1 requires erasing the temporal values of PIN, PUK. FIA_UAU.4/PACE, FIA_UAU.5/PACE and FCS_CKM.4 represent some required specific properties of the used protocols. To allow for a verification of the certificate chain as required in FMT_MTD.3, the CVCA's public key and certificate as well as the current date are written or update by authorized identified role as required by FMT_MTD.1.1/CVCA_INI, FMT_MTD.1.1/CVCA_UPD and FMT_MTD.1.1/DATE. Unauthorized modifying of the exchanged data is addressed by FTP_ITC.1/PACE in contactless using FCS_COP.1.1/SM in integrity. A prerequisite for establishing this trusted channel is a successful Chip Authentication, cf. FIA_API.1/TOE using FCS_CKM.1.1/DH_PACE possessing the special properties FIA_UAU.5/PACE and FIA_UAU.6/CA. As a prerequisite of this trusted channel a trusted channel established with the PACE protocol using FIA_UID.1/PACE, FIA_UAU.1/PACE and FCS_CKM.1.1/DH_PACE and possessing the special properties FIA_UAU.5/PACE, FIA_UAU.6/PACE. CA provides an evidence of possessing the Chip Authentication Private Key (CAPK). FMT_MTD.1.1/CAPK governs creating/loading CAPK, and FMT_MTD.1.1/KEY_READ requires CAPK to be unreadable by users; thus its value remains confidential. FDP_RIP.1 requires erasing the values of CAPK and session keys (here: for KMAC). The SFRs FMT_SMF.1 and FMT_SMR.1/PACE support related functions and roles.

This objective prevents TOE's functions being not intended to be used in the operational phase from manipulating and disclosing the User- and TSF-data. This objective is achieved by FMT_LIM.1 and FMT_LIM.2 preventing misuse of test and other functionality of the TOE having not to be used in the TOE's operational life cycle phase.

This objective protects against disclosure of confidential User- or/and TSF-data stored on / processed by the TOE. This objective is achieved

- by FPT_EMS.1 and FPT_EMS.1/PIN-PUK-KEYS for measurement and analysis of the shape and amplitude of signals or the time between events found by measuring signals on the electromagnetic field, power consumption, clock, or I/O lines,
- by FPT_FLS.1 and FPT_TST.1 for forcing a malfunction of the TOE, and
- by FPT_PHP.3 for a physical manipulation of the TOE.

This objective ensures a correct operation of the TOE by preventing its operation outside the normal operating conditions. This objective is covered by FPT_TST.1 requiring self tests to demonstrate the correct operation of the TOE and tests of authorised users to verify the integrity of the TSF-data and the embedded software (TSF code) as well as by FPT_FLS.1 requiring entering a secure state of the TOE in case of detected failure or operating conditions possibly causing a malfunction.

This objective protects of the confidentiality and integrity of the User- and TSF-data as well as embedded software stored in the TOE. This objective is completely covered by FPT_PHP.3 in an obvious way.

OT.Key_LifeCycle_Security (*Lifecycle security of the key(s) stored in the TOE*)
Secure Keys generation is ensured by **FCS_CKM.1/SCD/SVD_Generation**.

The secure keys usage is ensured cryptographically according to **FCS_COP.1/Digital Auth**, **FCS_COP.1/Enc key Decipherement**. Keys usage is based on the security attribute secure TSF management according to **FMT_MSA.2**, **FMT_MSA.3**, **FMT_SMF.1** and **FMT_SMR.1**. The test functions **FPT_TST.1** provides failure detection throughout the lifecycle.

The SFR FCS_CKM.4 ensures a secure keys destruction.

OT.Keys_Secrecy (Secrecy of key(s) stored in the TOE) is provided by the security functions specified by the following SFR.
FCS_CKM.1/SCD/SVD_Generation ensure the use of secure cryptographic algorithms for keys generation.

Cryptographic quality of the asymmetric key pair(s) shall prevent disclosure of the TOE's private authentication key(s) and eServices key(s) by cryptographic attacks using the publicly known public key.
The security functions specified by FDP_RIP.1 and FCS_CKM.4 ensure that residual information on a key(s) is destroyed after a key has been used for authentication (verification or proof) or an eServices keys has been used and that destruction of key(s) leaves no residual information.

The security functions specified by FDP_SDI.2/Persistent ensure that no critical data is modified which could alter the efficiency of the security functions or leak information of the authentication key. FPT_TST.1 tests the working conditions of the TOE and FPT_FLS.1 guarantees a secure state when integrity is violated and thus assures that the specified security functions are operational. An example where compromising error conditions are countered by FPT_FLS.1 is fault injection for differential fault analysis (DFA).

FPT_EMS.1/PIN-PUK-KEYS and FPT_PHP.3 require additional security features of the TOE to ensure the confidentiality of the key(s).

OT.TOE_AuthKey_Unique (Uniqueness of the TOE authentication key(s)) implements the requirement of practically unique TOE's authentication private key, which is provided by the cryptographic algorithms specified by FCS_CKM.1/SCD/SVD_Generation.

OT.Lifecycle_Management (Management of the TOE life cycle) ensures a correct separation of the TOE life cycle between phase 6 and 7.

In phase 6, FMT_MTD.1/TOE State ensures the TOE irreversibly switches from phase 6 to phase 7 under the sole control of the Personalization Agent. The Personalization Agent is authenticated with a mutual authentication performed with FCS_RND.1 and FCS_COP.1/GP and is authenticated with FMT_SMR.1.

In phase 7, FDP_ACC.1/Signature creation, FDP_ACC.1/SVD transfer, FDP_ACC.1/SCD/SVD_Generation, FDP_ACC.1/SCD import, FDP_ACF.1/Signature creation, FDP_ACF.1/SVD transfer, FDP_ACF.1/SCD/SVD_Generation, FDP_ACF.1/SCD import, FMT_MTD.1/Unblock_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MOF.1, FMT_MTD.1/Admin, FMT_MTD.1/Signatory ensures the Personalization Agent does not control the TOE anymore.
In phase 6, the Personalization Agent has complete control over the administrative functions of the TOE. It may import, erase, generate SCD/SVD, export SVD, manage Keys, create RAD and manage the configuration of the TOE as mandated in FMT_SMF.1, according to the security policies defined in FDP_ACC.1/SVD transfer, FDP_ACC.1/SCD/SVD_Generation, FDP_ACC.1/SCD import, FDP_ACF.1/SVD transfer, FDP_ACF.1/SCD/SVD_Generation, FDP_ACF.1/SCD import, It may as well change TOE State (FMT_MTD.1/TOE State). These functions are protected by the Personalization Agent authentication that cannot be bypassed to access these functions with the TSF specified by FIA_UID.1 and FIA_UAU.1. FMT_MSA.1/Admin, FMT_MSA.2, FMT_MSA.3 ensure that the sole Personalization Agent can realize these functions.

OT.eServices (Provision of eServices) is provided by the cryptographic mechanisms specified by (1) FCS_COP.1/Digital Auth, (2) FCS_COP.1/Enc key decipherment. These requirements ensure the cryptographic robustness of these eServices.
The eServices keys may be loaded, generated, and the matching public key may be exported as required by FMT_SMF.1 and FMT_SMR.1. These functions are protected by the proper Agent(s) authentication that cannot be bypassed to access these functions with the TSF specified by FIA_UID.1 and FIA_UAU.1. Effort to bypass the access control by a frontal exhaustive attack is blocked by FIA_AFL.1/RAD.

OT.AC_Pers_EAC ensures that only the personalization agent can write user- and TSF-Data into the TOE, and that some of this data cannot be altered after personalization. This property is covered by FDP_ACC.1/TRM and FDP_ACF.1/TRM requiring, amongst other, an appropriate authorization level of an Authentication Terminal. This authorization level can be achieved by terminal identification/authentication as required by the SFRs FIA_UID.1/PACE and FIA_UAU.1/PACE . The SFRs FMT_SMF.1 and FMT_SMR.1/PACE support the related functions and roles. Since only an Authentication Terminal can reach the necessary authorization level, using and managing the PIN (the related SFRs are FIA_AFL.1/AUTH, FMT_MTD.1.1/Resume_PIN, FMT_MTD.1/Change_PIN, FMT_MTD.1/Unblock_PIN, FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK) also support the achievement of this objective. FDP_RIP.1 requires erasing the temporal values PIN and PUK. Finally, FMT_MTD.1.1/KEY_READ ensures that cryptographic keys for EAC cannot be read by users.

OT.Tracing ensures that the TOE prevents gathering TOE tracing data by means of unambiguously identifying the electronic document remotely through establishing or listening to communication via the contactless-based interface of the TOE without a priori knowledge of the correct values of shared passwords (CAN, PIN, PUK). This objective is achieved by FIA_AFL.1/AUTH and FTP_ITC.1/PACE.

8.3.3 Satisfaction of dependencies of security requirements

8.3.3.1 Dependencies

Functional Requirement	Dependencies	Satisfied by
FCS_CKM.1/SCD/SVD_Generation	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1/Sign FCS_CKM.4
FCS_CKM.1/Session Keys	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1/SM in confidentiality FCS_COP.1/SM in integrity FCS_CKM.4
FCS_CKM.1.1/DH_PACE	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1/SM in confidentiality FCS_COP.1/SM in integrity FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.1/SCD FCS_CKM.1/SCD/SVD_Generation
FCS_COP.1/Sign	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/SCD FCS_CKM.1/SCD/SVD_Generation FCS_CKM.4
FCS_COP.1/SM in confidentiality	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_CKM.1/Session Keys FDP_CKM.4/Session Keys
FCS_COP.1/SM in integrity	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_CKM.1/Session Keys FDP_CKM.4/Session Keys
FCS_COP.1/Digital Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1/SCD/SVD_Generation FCS_CKM.4
FCS_COP.1/Enc key decipherment	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1/SCD/SVD_Generation FCS_CKM.4
FCS_COP.1/GP secret data protection	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1/Session Keys FCS_CKM.4
FCS_COP.1/SIG_VER	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1.1/DH_PACE, FCS_CKM.4
FCS_RND.1	No dependencies	n/a
FDP_ACC.1/SCD/SVD_Generation	FDP_ACF.1	FDP_ACF.1/SCD/SVD_Generation
FDP_ACC.1/SCD_Import	FDP_ACF.1	FDP_ACF.1/SCD_Import
FDP_ACC.1/SVD_Transfer	FDP_ACF.1	FDP_ACF.1/SVD_Transfer
FDP_ACC.1/Signature_Creation	FDP_ACF.1	FDP_ACF.1/Signature_Creation
FDP_ACC.1/TRM	FDP_ACF.1	FDP_ACF.1/TRM
FDP_ACF.1/SCD/SVD_Generation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/SCD/SVD_Generation FMT_MSA.3
FDP_ACF.1/SCD_Import	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/SCD_Import FMT_MSA.3
FDP_ACF.1/SVD_Transfer	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/SVD_Transfer FMT_MSA.3
FDP_ACF.1/Signature_Creation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signature_Creation FMT_MSA.3
FDP_ACF.1/TRM	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/TRM FMT_MSA.3
FDP_RIP.1	No dependencies	n/a
FDP_SDI.2/Persistent	No dependencies	n/a
FDP_SDI.2/DTBS	No dependencies	n/a
FDP_ITC.1/SCD	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	FDP_ACC.1/SCD_Import FMT_MSA.3
FDP_UCT.1/SCD	[FTP_ITC.1 or FTP_TRP.1], [FDP_ACC.1 or FDP_IFC.1]	FTP_ITC.1/SCD FDP_ACC.1/SCD_Import

Functional Requirement	Dependencies	Satisfied by
FDP_UCT.1/TRM	[FTP_ITC.1 or FTP_TRP.1], [FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1/TRM, FTP_ITC.1/PACE
FDP_UIT.1/TRM	[FTP_ITC.1 or FTP_TRP.1], [FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1/TRM, FTP_ITC.1/PACE
FDP_DAU.2/SVD	FIA_UID.1	FIA_UID.1
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.1/PACE	FIA_UID.1	FIA_UID.1/PACE
FIA_UAU.4/PACE	No Dependencies	n/a
FIA_UAU.5/PACE	No Dependencies	n/a
FIA_UAU.6/PACE	No Dependencies	n/a
FIA_UAU.6/CA	No Dependencies	n/a
FIA_UID.1	No dependencies	n/a
FIA_UID.1/PACE	No dependencies	n/a
FIA_AFL.1/RAD	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/AUTH	FIA_UAU.1	FIA_UAU.1/PACE
FIA_API.1	No dependencies	n/a
FIA_API.1/TOE	No dependencies	n/a
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_SMR.1/PACE	FIA_UID.1	FIA_UID.1/PACE
FMT_SMF.1	No dependencies	n/a
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MSA.1/Admin	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	FDP_ACC.1/SCD/SVD_Generation, FDP_ACC.1/SCD_Import FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/Signatory	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	FDP_ACC.1/Signature_Creation, FMT_SMR.1, FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1], FMT_MSA.1, FMT_SMR.1	FDP_ACC.1/SCD/SVD_Generation, FDP_ACC.1/SCD_Import FDP_ACC.1/Signature_Creation, FMT_MSA.1/Admin, FMT_MSA.1/Signatory FMT_SMR.1,
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/Admin, FMT_MSA.1/Signatory, FMT_SMR.1
FMT_MSA.4	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1/SCD/SVD_Generation, FDP_ACC.1/SCD_Import, FDP_ACC.1/Signature_Creation
FMT_MTD.1/Admin	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/Signatory	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1.1/CVCA_INI	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/CVCA_UPD	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/DATE	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/CAPK	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/KEY_READ	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/Initialize_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/Resume_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1/Change_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1/Unblock_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/UnblockChange_RAD	FMT_SMR.1,	FMT_SMR.1/PACE, FMT_SMF.1

Functional Requirement	Dependencies	Satisfied by
	FMT_SMF.1	
FMT_MTD.1.1/Erase_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/Reinitialize_PIN	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/UnblockChange_PUK	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1/PACE, FMT_SMF.1
FMT_MTD.1.1/TOE state	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.3	(FMT_MTD.1)	FMT_MTD.1/CVCA_INI, FMT_MTD.1/CVCA_UPD, FMT_MTD.1/DATE
FMT_LIM.1	No dependencies	n/a
FMT_LIM.2	No dependencies	n/a
FPT_EMS.1	No dependencies	n/a
FPT_EMS.1/PIN-PUK-KEYS	No dependencies	n/a
FPT_FLS.1	No dependencies	n/a
FPT_PHP.1	No dependencies	n/a
FPT_PHP.3	No dependencies	n/a
FPT_TST.1	No dependencies	n/a
FTP_ITC.1/SCD	No dependencies	n/a
FTP_ITC.1/SVD	No dependencies	n/a
FTP_ITC.1/PACE	No dependencies	n/a

Table 7 - Satisfaction of dependencies of SFR

Assurance Requirement	Dependencies	Satisfied by
EAL5 package	(dependencies of EAL5 package are not reproduced here)	By construction, all dependencies are satisfied in a CC EAL package
AVA_VAN.5	ADV_ARC.1 ADV_FSP.4 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1	ADV_ARC.1 ADV_FSP.4 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1 (all are included in EAL5 package)
ALC_DVS.2	No dependencies	n/a

Table 8 - Satisfaction of dependencies of SAR

9 TOE summary Specifications

9.1 Description

The TOE inherits all the security functions provided by the underlying javacard open platform [PLT] (see the Security target). On top of these, it adds some supplemental security functions that are described hereafter.

9.1.1 SF.PIN_MGT

This security function is involved in the management of the PINs (PIN, PUK and RAD). PINs are secret data used so that a natural person can authenticate itself to the TOE.
This security function (1) provides all management operations on these PINs (verification, change, unblocking, unblocking and change, initialization, re initialization) and (2) enforces the access control policies over these operations.

The natural person can authenticate itself to the TOE via the PACE protocol and/or the VERIFY PIN command. Notice that The usage of PACE protocol is mandatory only for contactless mode. The verification process uses a velocity checking mechanism, thus a remaining tries counter and a maximum error counter are defined for each PIN. If the verification fails, the tries counter is decremented by one and an error status that contains the remaining attempts is returned by the application. When all available tries have failed, the PIN is suspended or blocked and can no longer be used. Note that a successful verification of the PIN resets its remaining tries counter to the maximum error counter.

9.1.2 SF.SIG

This security function manages the signature creation service.

It enforces access control over the signature creation service:

- In phase 6, it ensures the signature computation function is not accessible, and in particular that the personalization agent cannot sign on behalf of the Signatory.
- In phase 7, it ensures the signature creation feature is activated only by the signatory.
- In phase 7, it enforces the integrity of DTBS, and ensures that R.Sigy is successfully authenticated before creating the signature.

The security function ensures the data hashing (if hash on card, or partial hashing is used), and the secure signature computation using the private key (SCD), either with RSA or EC-DSA cryptography.

This security function relies on SF.PIN_MGT to authenticate the Signatory.

9.1.3 SF. AUTH

This security function manages the authentication protocols supported by the TOE.

This security function supports the authentication with the personalization agent in phase 6. This authentication relies on a mutual authentication protocol authenticating (1) the TOE to the personalization agent, and (2) the personalization agent to the TOE. It relies on symmetric master key sets shared by the TOE and the personalization agent that may be DES or AES symmetric keys (depending on the type of Secure Channel Protocol – SCP). Upon successful completion of the authentication, both parties (TOE and personalization agent) generate session keys that may be used to establish a secure channel thanks to SF.SM. This secure channel allows protecting the communication between them in integrity, authenticity and confidentiality. Each symmetric master key sets is associated to an error counter, which aims at ensuring its protecting against brute force attacks. Upon each wrong authentication attempt, the authentication processing time is slowed down (increasingly). Once a successful authentication takes place, the slow down mechanisms is cancelled.

In phase 7, this security function also supports the Extended Access Control protocol (EAC) made up with Chip authentication (CA) and Terminal Authentication (TA). This protocol allows (1) a mutual authentication between the TOE and a remote terminal, and (2) the generation of session keys used to establish a secure channel thanks to SF.SM. This secure channel protects the communication between them in integrity, authenticity and confidentiality. This authentication is based on PKI scheme : each party (TOE and remote terminal) uses (1) an authentication private key and (2) digital certificates containing the corresponding authentication public key and linking it to the root of trust known by the other party, to authenticate itself to the other party. This security function manages the verification and processing of the certificates received from the remote terminal (that have a specific format named Card Verifiable Certificate – CVC). In particular, this security function computes the effective authorization of the remote terminal.
The Extended Access Control (EAC) protocol is used to establish a trusted channel with the CGA prior to SCD/SVD generation.

In phase 7, this security function also supports the PACE protocol. This protocol is a human to machine authentication protocol allowing to (1) authenticate a natural person to the TOE by using a PIN or (2) prove the holder has the TOE in hand (using CAN) and (3) creating a secure channel between the TOE and a device used to initiate the communication. This protocol is designed to protect the secrecy of the PIN in the course of the authentication so that it can't be intercepted during the communication, or deduced through crypto analysis. Upon successful completion of PACE authentication, session keys used to establish a secure channel thanks to SF.SM are generated. This secure function also manage error counter on the credentials used to perform the authentication.

- Upon failure, the error counter of the credentials used to initiate the PACE authentication is decreased until it is blocked.
- Once blocked, a secret credential (PINs) can't be used anymore to perform a PACE protocol, while for CAN, the PACE protocol is slowed down.
- Upon successful PACE authentication, the error counter of the credentials used to initiate the PACE authentication is reset to its maximum value;

Last but not least, this security function also handles the suspension mechanisms protecting the PINs against Denial Of services attacks (not applicable to CAN). When the remaining number of tries is set to '1', any attempt to perform a PACE authentication in contactless mode using a PIN credential will require it to be made through a secure channel generated using PACE authentication performed using the CAN.

In phase 7, this security function also provides mechanisms to authenticate the SSCD and prove its identity (thanks to the Chip Authentication mechanism described above).

9.1.4 SF.SM

This security function ensures the protection of communication between the TOE and an external entity. As such, this security function maintains a trusted channel between the TOE and an external entity.

This security function requires the TOE and the external entity to establish first a trusted channel using an authentication supported by SF.AUTH.

This security function ensures the following properties:

- In phase 6, it ensures the confidentiality, integrity and authenticity of the private keys (including the SCD), and the PINs (including the RAD);
- In phase 6, it ensures the integrity and authenticity of the asymmetric public key (including the SVD) when being exported to the outside
- In phase 7, it ensures the confidentiality, integrity and authenticity of communication between the TOE and the external entity with which an authentication was performed;

In phase 6, the confidentiality, integrity and authenticity of communication is ensured by symmetric cryptography. Sensitive data (such as SCD or PINs) are encrypted using a dedicated symmetric session keys for data encryption generated from the seed agreed during the authentication with the personalization agent (see SF.AUTH). On top of that, data are encrypted and signed using symmetric session keys generated from the seed agreed during the authentication with the personalization agent (see SF.AUTH).

In phase 7, the confidentiality, integrity and authenticity of communication is ensured by symmetric cryptography. Data are encrypted and signed using symmetric session keys generated from the seed agreed during the Extended Access

Control (EAC) or PACE protocol (see SF.AUTH). Moreover, the protection against replay attacks is ensured by the Message Authentication Code (MAC) which is computed using a dynamic ICV, incremented at each new command.

This security function is also in charge of:

- generating the session keys from the seed computed by SF.AUTH. These session keys are ephemeral and unique, as the seed is computed from random numbers generated by the TOE and the external entity.
- destroying the session keys in case an error is detected (data not authentic or not integer), or when a command in plain text is sent.
- providing CGA the ability to verify evidence of the validity of the SVD and the identity of the user that generated the evidence.

This security function supports SF.AUTH to prove the identity of the SSCD and the TOE holder.

9.1.5 SF.KEY_MGT

This security function is involved in the management of the asymmetric keys (including SCDs and SVDs).

It enforces access control over any management operation on the keys:

- **In phase 6**, it only allows the key (including the SCD) to be loaded, generated and exported (for the public keys) by the personalization agent. It also requires the private keys to be encrypted in order to ensure their confidentiality. This security function ensures the personalization agent (1) can't use the keys it has loaded or generated. and (2) can't impersonate the associated role (in case of authentication keys), or create a signature with the SCD;
- **In phase 7**, it allows managing all the keys (including the SCD) by providing generation and export of the corresponding public key. It also enforces access control policies on these operations.

This security function also ensures that after update or generation, the former key (including SCD and SVD) is securely destroyed.

9.1.6 SF.CONF

This security function manages the configuration of the TOE in phase 6. For instance it allows the modification of the TOE State in phase 6.

This security function ensures an access control over these operations. Only the successfully authenticated Personalization Agent can modify these attributes.

This security function relies on SF.AUTH to authenticate the personalization agent.

9.1.7 SF.ESERVICE

This security function enables to perform digital authentication and electronic services. It is active in phase 7.

This security function offers the following services:

- Digital authentication;
- Decryption key decipherment;

This security function relies on SF.KEY_MGT which provides management of the keys on which these services rely.

9.1.8 SF.SAFESTATE_MGT

This security function ensures the TOE is always in a safe state. It monitors the integrity of the TOE, its assets and the TSF data by performing self-tests. When an unexpected event occurs (loss of power, loss of integrity, tearing,…), it ensures

- the TOE returns in a safe state
- all sensitive data are erased

- the TOE returns in a restrictive secure state

When a major issue is detected, the security function ensures the destruction of the TOE, so that the assets are not accessible anymore.

9.1.9 SF.PHYS

This security function protects the TOE against physical attacks, so that the integrity and confidentiality of the TOE is ensured, including keys, user data, CPLC data, configuration data and TOE life cycle. It detects physical tampering, responds automatically and also controls the emanations sent out by the TOE. It furthermore prevents deploying test features after TOE delivery.

9.2 SFRs and TSS

9.2.1 SFRs and TSS – Rationale

- SF.PIN_MGT: The implementation of this security function contributes to:

  - FIA_UID.1, FIA_UAU.1, FIA_UID.1/PACE, FIA_UAU.1/PACE, FIA_UAU.4/PACE, FIA_UAU.5/PACE that provide user identification and user authentication prior to enabling access to authorized functions.

  - FIA_AFL.1/RAD, FIA_AFL.1/AUTH that handle the authentication failure.

  - FMT_SMR.1, FMT_SMR.1/PACE that define security roles for the TOE.

  - FMT_SMF.1.

  - FTP_ITC.1/PACE that ensures a trusted channel between them TOE and a PACE terminal to protect the exchanged data in contactless from modification and disclosure.

  - FMT_MTD.1/Admin, FMT_MTD.1/Signatory, FMT_MTD.1.1/Initialize_PIN, FMT_MTD.1.1/Resume_PIN,FMT_MTD.1/Change_PIN,FMT_MTD.1/Unblock_PIN,FMT_MTD.1.1/UnblockChange_RAD, FMT_MTD.1.1/Erase_PIN, FMT_MTD.1.1/Reinitialize_PIN, FMT_MTD.1.1/UnblockChange_PUK, FMT_MTD.3 that manage the TSFs date by defining access rules.

- SF.SIG: The implementation of this security function contributes to:
  - FCS_COP.1/Sign that provide cryptographic operations.
  - FDP_ACC.1/Signature_Creation, FDP_ACF.1/Signature_Creation that enforce the signature creation SFP.
  - FIA_UID.1, FIA_UAU.1 that provide user identification and user authentication prior to enabling access to autorized functions.
  - FMT_MSA.1/Signatory, FMT_MSA.2, FMT_MSA.3, FMT_MSA.4 that manages the access right policy of the TOE.
  - FMT_MOF.1 that ensures the management of the signature creation function.
  - FMT_SMF.1.
  - FDP_SDI.2/DTBS that ensures the integrity of DTBS.

- SF.AUTH: The implementation of this security function contributes to:
  - FCS_CKM.1.1/DH_PACE that ensures cryptographic key generation.
  - FCS_COP.1/SM in confidentiality, FCS_COP.1/SM in integrity, FCS_COP.1/GP secret data protection, FCS_RND.1, FCS_COP.1.1/SIG_VER that provide cryptographic operations.
  - FIA_API.1, FIA_API.1/TOE
  - FMT_SMR.1, FMT_SMR.1/PACE that define security roles for the TOE.
  - FMT_SMF.1
  - FMT_MSA.1/Admin, FMT_MSA.2, FMT_MSA.3, FMT_MSA.4 that manages the access right policy of the TOE.
  - FTP_ITC.1/SCD, FTP_ITC.1/SVD, FTP_ITC.1/PACE that ensure a trusted channel between them TOE and a Terminal to protect the exchanged data from modification and disclosure.
  - FIA_UID.1, FIA_UAU.1, FIA_UID.1/PACE, FIA_UAU.1/PACE, FIA_UAU.4/PACE, FIA_UAU.5/PACE, FIA_UAU.6/PACE, FIA_UAU.6/CA that provide user identification and user authentication prior to enabling access to authorized functions.
  - FIA_AFL.1/AUTH that handles the authentication failure.
  - FDP_ACC.1/TRM, FDP_ACF.1/TRM that ensure that only an authorized user under specific conditions can perform a dedicated operation, and in particular, that only users authenticated as authenticated terminal.

- FDP_UCT.1/TRM that ensures data exchange confidentiality.

- FMT_MTD.1.1/CVCA_INI, FMT_MTD.1.1/CVCA_UPD, FMT_MTD.1.1/DATE, FMT_MTD.1.1/CAPK, FMT_MTD.1.1/KEY_READ, FMT_MTD.3 that manage the TSFs date by defining access rules.

o SF.SM: The implementation of this security function contributes to:

- FCS_CKM.1/Session Keys that ensure cryptographic key generation.

- FCS_COP.1/SM in confidentiality, FCS_COP.1/SM in integrity, FCS_COP.1/GP secret data protection, FCS_COP.1.1/SIG_VER that provide cryptographic operations.

- FDP_DAU.2/SVD that ensures that exported SVD to the CGA is authenticated and unmodified.

- FIA_API.1, FIA_API.1/TOE.

- FTP_ITC.1/SCD, FTP_ITC.1/SVD, FTP_ITC.1/PACE that ensure a trusted channel between them TOE and a Terminal to protect the exchanged data from modification and disclosure.

- FIA_UID.1/PACE, FIA_UAU.1/PACE, FIA_UAU.4/PACE, FIA_UAU.5/PACE,FIA_UAU.6/PACE, FIA_UAU.6/CA that provide user identification and user authentication prior to enabling access to authorized functions.

- FIA_AFL.1/AUTH.

- FDP_ACC.1/TRM, FDP_ACF.1/TRM that ensure that only an authorized user under specific conditions can perform a dedicated operation, and in particular, that only users authenticated as authenticated terminal.

- FDP_UCT.1/SCD, FDP_UCT.1/TRM that ensures data exchange confidentiality.

- FDP_UIT.1/TRM that ensures data exchange integrity.

o SF.KEY_MGT: The implementation of this security function contributes to:

- FCS_CKM.1/SCD/SVD_Generation that ensures cryptographic key generation.

- FCS_CKM.4 that manages that ensure cryptographic key destruction.

- FDP_ACC.1/SCD/SVD_Generation,FDP_ACF.1/SCD/SVD_Generation,FDP_ACC.1/SVD_Transfer, FDP_ACF.1/SVD_Transfer, FDP_ACC.1/SCD_Import, FDP_ACF.1/SCD_Import that ensure that only an authorized user under specific conditions can perform a dedicated operation, and in particular, that only users authenticated as administrator or signatory.

- FDP_ITC.1/SCD that ensures a trusted channel between them TOE and a Terminal to protect the exchanged data from modification and disclosure.

- FIA_UID.1, FIA_UAU.1 that provide user identification and user authentication prior to enabling access to authorized functions.

- FMT_SMF.1.

- FMT_MSA.2, FMT_MSA.3, FMT_MSA.4 that manage the access right policy of the TOE.
- FDP_UCT.1/TRM that ensures data exchange confidentiality.
- FDP_UIT.1/TRM that ensures data exchange integrity.

o SF.CONF: The implementation of this security function contributes to:

- FMT_SMF.1.
- FMT_MSA.2, FMT_MSA.3 that manage the access right policy of the TOE.
- FMT_MTD.1/TOE state that restrict the ability to switch the TOE from phase 6 to phase 7 to the personalization agent.

o SF.ESERVICE: The implementation of this security function contributes to:

- FCS_COP.1/Digital Auth, FCS_COP.1/Enc key decipherment that provide cryptographic operations.

o SF.SAFESTATE_MGT: The implementation of this security function contributes to:

- FDP_RIP.1 that ensures erasure of data in FLASH and in RAM.
- FDP_SDI.2/Persistent, FDP_SDI.2/DTBS that ensure the integrity of data stored in the TOE.
- FPT_FLS.1 that ensure the preservation of secure state when failures occur.
- FPT_TST.1 that ensures the integrity of the data stored on the TOE.

o SF.PHYS: The implementation of this security function contributes to:

- FPT_EMS.1, FPT_EMS.1/PIN-PUK-KEYS that ensure the TOE emanation.
- FPT_PHP.1, FPT_PHP.3 that ensures the detection of physical tampering of the TOE and the resistance to it.
- FMT_LIM.1, FMT_LIM.2.

9.2.2 Matrix coverage

SFR \ SF	SF.PIN_MGT	SF.SIG	SF.AUTH	SF.SM	SF.KEY_MGT	SF.CONF	SF.ESERVICE	SF.SAFESTATE_MGT	SF.PHYS
FCS_CKM.1/SCD/SVD_Generation					X				
FCS_CKM.1/Session Keys				X					
FCS_CKM.4					X				
FCS_COP.1/Sign		X							
FCS_COP.1/SM in confidentiality			X	X					
FCS_COP.1/SM in integrity			X	X					
FCS_COP.1/Digital Auth							X		
FCS_COP.1/Enc key decipherment							X		
FCS_COP.1/GP secret data protection			X	X					
FCS_RND.1			X						
FDP_ACC.1/SCD/SVD_Generation					X				
FDP_ACC.1/SCD_Import					X				
FDP_ACC.1/SVD_Transfer					X				
FDP_ACC.1/Signature_Creation		X							
FDP_ACF.1/SCD/SVD_Generation					X				
FDP_ACF.1/SCD_Import					X				
FDP_ACF.1/SVD_Transfer					X				
FDP_ACF.1/Signature_Creation		X							
FDP_RIP.1								X	
FDP_SDI.2/Persistent								X	
FDP_SDI.2/DTBS		X						X	
FDP_ITC.1/SCD					X				
FDP_UCT.1/SCD				X					
FDP_DAU.2/SVD				X					
FIA_UAU.1	X	X	X		X				
FIA_UID.1	X	X	X		X				
FIA_AFL.1/RAD	X								
FIA_API.1			X	X					
FMT_SMR.1	X		X						
FMT_SMF.1	X	X	X		X	X			
FMT_MOF.1		X							
FMT_MSA.1/Admin			X						
FMT_MSA.1/Signatory		X							
FMT_MSA.2		X	X		X	X			
FMT_MSA.3		X	X		X	X			
FMT_MSA.4		X	X		X				
FMT_MTD.1/Admin	X								
FMT_MTD.1/Signatory	X								
FMT_MTD.1/TOE state						X			
FPT_EMS.1									X
FPT_FLS.1								X	
FPT_PHP.1									X
FPT_PHP.3									X
FPT_TST.1								X	
FTP_ITC.1/SCD			X	X					
FTP_ITC.1/SVD			X	X					
FCS_CKM.1.1/DH_PACE			X						
FCS_COP.1.1/SIG_VER			X	X					
FIA_UAU.1/PACE	X		X	X					
FIA_UAU.5/PACE	X		X	X					
FIA_AFL.1/AUTH	X		X	X					
FIA_API.1/TOE			X	X					
FIA_UAU.6/CA			X	X					
FIA_UID.1/PACE	X		X	X					
FIA_UAU.4/PACE	X		X	X					

SFR \ SF	SF.PIN_MGT	SF.SIG	SF.AUTH	SF.SM	SF.KEY_MGT	SF.CONF	SF.ESERVICE	SF.SAFESTATE_MGT	SF.PHYS
FIA_UAU.6/PACE			X	X					
FDP_ACF.1/TRM			X	X					
FDP_ACC.1/TRM			X	X					
FDP_UCT.1/TRM			X	X	X				
FDP_UIT.1/TRM				X	X				
FTP_ITC.1/PACE	X		X	X					
FMT_SMR.1/PACE	X		X						
FMT_MTD.1.1/CVCA_INI			X						
FMT_MTD.1.1/CVCA_UPD			X						
FMT_MTD.1.1/DATE			X						
FMT_MTD.1.1/CAPK			X						
FMT_MTD.1.1/KEY_READ			X						
FMT_MTD.1.1/Initialize_PIN	X								
FMT_MTD.1.1/Resume_PIN	X								
FMT_MTD.1/Change_PIN	X								
FMT_MTD.1/Unblock_PIN	X								
FMT_MTD.1.1/UnblockChange_RAD	X								
FMT_MTD.1.1/Erase_PIN	X								
FMT_MTD.1.1/Reinitialize_PIN	X								
FMT_MTD.1.1/UnblockChange_PUK	X								
FMT_MTD.3	X		X						
FMT_LIM.1									X
FMT_LIM.2									X
FPT_EMS.1/PIN-PUK-KEYS									X