IDEMIA

IAS ECC V2, version 1.3,

in configuration #2

on ID-One Cosmo v8.2

open platform

on NXP P6022M VB

Public Security Target

About IDEMIA

OT-Morpho is now IDEMIA, the global leader in trusted identities for an increasingly digital world, with the ambition to empower citizens and consumers alike to interact, pay, connect, travel and vote in ways that are now possible in a connected environment.

Securing our identity has become mission critical in the world we live in today. By standing for Augmented Identity, we reinvent the way we think, produce, use and protect this asset, whether for individuals or for objects. We ensure privacy and trust as well as guarantee secure, authenticated and verifiable transactions for international clients from Financial, Telecom, Identity, Security and IoT sectors.

With close to €3bn in revenues, IDEMIA is the result of the merger between OT (Oberthur Technologies) and Safran Identity & Security (Morpho). This new company counts 14,000 employees of more than 80 nationalities and serves clients in 180 countries.

For more information, visit www.idemia.com / Follow @IdemiaGroup on Twitter

IDEMIA

DOCUMENT MANAGEMENT

Business Unit – Department	**PSI**
Document type	**FQR**
Document Title	**IAS ECC V2 in configuration #2 on Cosmo V8.2 - Public Security Target**
FQR No	**110 9185**
FQR Issue	**5**
Project Name	**IAS ECC V2**

DOCUMENT REVISION

Date	Revision	Modification	Modified by
13/05/2019	1.0	Creation	**IDEMIA**
29/07/2019	2.0	Add Underlying Platform Certification Reference	**IDEMIA**
21/04/2020	3.0	Added updated platform certificate	**IDEMIA**
06/07/2023	4.0	Updates for reevaluation	IDEMIA
22/09/2023	**5**.0	Updates: add QR guidance reference	IDEMIA

IDEMIA

TABLE OF CONTENTS

1 DEFINITIONS 7

2 REFERENCES 8

3 SECURITY TARGET INTRODUCTION 10

3.1 PUBLIC SECURITY TARGET REFERENCE 10
3.2 TOE REFERENCE 10
3.3 TOE OVERVIEW 11
3.3.1 TOE Type 11
3.3.2 Logical scope 11
3.3.3 Physical scope 12
3.3.4 Required non-TOE hardware/software/firmware 12
3.3.5 Usage and major security features 12
3.3.6 Scope of evaluation 13
3.4 TOE DESCRIPTION 13
3.4.1 Data structure 13
3.4.1.1 File and File System 13
3.4.1.2 Security Environment 15
3.4.1.3 Security data Objects 15
3.4.2 Access Control Management 16
3.4.3 Authentication of entities 16
3.4.4 Electronic Services 16
3.4.5 Administration of the TOE 16
3.4.6 Single Sign on feature (SSO) 17
3.5 LIFE CYCLE 17
3.5.1 Development 18
3.5.1.1 Software development (phase 1) 18
3.5.1.2 Hardware development (Phase 2) 18
3.5.1.3 Javacard open platform manufacturing (phase 3) 18
3.5.2 Production 19
3.5.2.1 Packaging and initialization (phase 4) 19
3.5.2.2 Preparation (phase 5) 19
3.5.3 Operational state 19
3.5.3.1 Applet pre-personalization (phase 6) 19
3.5.3.2 TOE personalization (phase 6) 20
3.5.3.3 TOE Usage (phase 7) 20
3.5.4 Coverage of the different Life cycle state by the assurance components [AGD] & [ALC] 20
3.5.5 State of the TOE depending on the phase 21
3.5.6 Mapping with the Users 21

4 CONFORMANCE CLAIM 22

4.1 CC AND PACKAGE CONFORMANCE CLAIM 22
4.2 PP CONFORMANCE CLAIM 22
4.3 CONFORMANCE RATIONALE 22
4.3.1 Life cycle conformance 22

4.3.2 Additional assets 23
4.3.3 Additional Roles 23
4.3.4 Additional threats 23
4.3.5 Additional OSPs 23
4.3.6 Additional objectives 24
4.3.6.1 Additional Security objectives for the TOE 24
4.3.6.2 Additional Security objectives for the Operational Environment 24
4.3.7 Additional SFRs 24
4.3.8 Package conformance 25
5 SECURITY PROBLEM DEFINITION 25
5.1 ASSETS AND USERS 25
5.1.1 Assets 25
5.1.1.1 Assets from protection profiles: User Data 25
5.1.1.2 Additional Assets : TSF Data 25
5.1.2 Subjects 26
5.2 THREATS 26
5.2.1 Threats drawn from the protection profiles 26
5.2.1.1 T.SCD_Divulg *Storing, copying and releasing of the signature creation data* 26
5.2.1.2 T.SCD_Derive *Derive the signature creation data* 26
5.2.1.3 T.Hack_Phys *Physical attacks through the TOE interfaces* 26
5.2.1.4 T.SVD_Forgery *Forgery of the signature verification data* 26
5.2.1.5 T.SigF_Misuse *Misuse of the signature creation function of the TOE* 26
5.2.1.6 T.DTBS_Forgery *Forgery of the DTBS/R* 26
5.2.1.7 T.Sig_Forgery *Forgery of the electronic signature* 27
5.2.2 Additional threats 27
5.2.2.1 T.Key_Divulg *Storing, copying, and releasing of a key stored in the TOE* 27
5.2.2.2 T.Key_Derive *Derive a key* 27
5.2.2.3 T.TOE_PublicAuthKey_Forgery *Forgery of the public key of a TOE authentication key* 27
5.2.2.4 T.Authentication_Replay *Replay of an authentication of an external entity* 27
5.3 ORGANISATIONAL SECURITY POLICIES 27
5.3.1 Security policies drawn from the protection profiles 27
5.3.1.1 P.CSP_QCert *Qualified certificate* 27
5.3.1.2 P.Qsign *Qualified electronic signatures* 27
5.3.1.3 P. Sigy_SSCD *TOE as secure signature creation device* 27
5.3.1.4 P.Sig_Non-Repud *Non-repudiation of signatures* 28
5.3.2 Additional security policies 28
5.3.2.1 P.LinkSCD_QualifiedCertificate *Link between a SCD stored in the TOE and the relevant qualified certificate* 28
5.3.2.2 P.TOE_PublicAuthKey_Cert *Certificate for asymmetric TOE authentication keys* 28
5.3.2.3 P.TOE_Construction *Construction of the TOE by the Personalization Agent* 28
5.3.2.4 P.eServices Provision of eServices 28
5.4 ASSUMPTIONS 28
5.4.1 A.CGA Trustworthy certificate generation application 28

5.4.2 A.SCA Trustworthy signature creation application 28
5.4.3 A.CSP Secure SCD/SVD management by SCD 28
6 SECURITY OBJECTIVES 29
6.1 SECURITY OBJECTIVES FOR THE TOE 29
6.1.1 Security Objectives drawn from the protection profiles 29
6.1.1.1 OT.Lifecycle_Security *Lifecycle security* 29
6.1.1.2 OT.SCD/SVD_Auth_Gen *Authorized SCD/SVD generation* 29
6.1.1.3 OT.SCD_Unique *Uniqueness of the signature creation data* 29
6.1.1.4 OT.SCD_SVD_Corresp *Correspondence between SVD and SCD* 29
6.1.1.5 OT.SCD_Auth_Imp *Authorized SCD import* 29
6.1.1.6 OT.SCD_Secrecy *Secrecy of the signature creation data* 29
6.1.1.7 OT.Sig_Secure *Cryptographic security of the electronic signature* 29
6.1.1.8 OT.Sigy_SigF *Signature creation function for the legitimate signatory only* 29
6.1.1.9 OT.DTBS_Integrity_TOE *DTBS/R integrity inside the TOE* 29
6.1.1.10 OT.EMSEC_Design *Provide physical emanations security* 29
6.1.1.11 OT.Tamper_ID *Tamper detection* 30
6.1.1.12 OT.Tamper_Resistance *Tamper resistance* 30
6.1.2 Additional Security Objectives for the TOE 30
6.1.2.1 OT.Authentication_Secure *Secure authentication mechanisms* 30
6.1.2.2 OT.SCD/SVD_Management *Management of SCD/SVD* 30
6.1.2.3 OT.Key_Lifecycle_Security *Life cycle security of the keys stored in the TOE* 30
6.1.2.4 OT.Keys_Secrecy *Secrecy of Keys* 30
6.1.2.5 OT.TOE_AuthKey_Unique *Uniqueness of the TOE authentication key(s)* 30
6.1.2.6 OT.Lifecycle_Management *Management of the life cycle* 30
6.1.2.7 OT.eServices *Provision of eServices* 31
6.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 31
6.2.1 Security Objectives drawn from the protection profiles 31
6.2.1.1 OE.SVD_Auth *Authenticity of the SVD* 31
6.2.1.2 OE.CGA_QCert *Generation of qualified certificates* 31
6.2.1.3 OE.SSCD_Prov_Service *Authentic SSCD provided by SSCD Provisioning Service* 31
6.2.1.4 OE.HID_VAD *Protection of the VAD* 31
6.2.1.5 OE.DTBS_Intend *SCA sends data intended to be signed* 31
6.2.1.6 OE.DTBS_Protect *SCA protects the data intended to be signed* 31
6.2.1.7 OE.Signatory *Security obligation of the signatory* 31
6.2.1.8 OE.SCD/SVD_Auth_Gen *Authorized SCD/SVD generation* 32
6.2.1.9 OE.SCD_Secrecy *SCD Secrecy* 32
6.2.1.10 OE.SCD_Unique *Uniqueness of the signature creation data* 32
6.2.1.11 OE.SCD_SVD_Corresp *Correspondance between SVD and SCD* 32
6.2.2 Additional security objectives for the operational environment 32
6.2.2.1 OE.LinkSCD_QualifiedCertificate *Link betwaan SCD stored in the TOE and the relevant qualified certificate* 32
6.2.2.2 OE.AuthKey_Transfer *Secure transfer of authentication key(s) to the TOE* 32

6.2.2.3 OE.AuthKey_Unique *Uniqueness of the authentication key(s)* 32
6.2.2.4 OE.TOE_PublicKeyAuth_Transfer *Secure transfer of public authentication key(s) of the TOE* 32
6.2.2.5 OE_TOE_Construction *Construction of the TOE by the Personalisation_Agent* 32
6.3 SECURITY OBJECTIVES RATIONALE 34
6.3.1 Security objectives backtracking 34
6.3.2 Security objectives sufficiency 36
7 EXTENDED COMPONENTS DEFINITION 39
7.1 FPT_EMS TOE EMANATION 39
7.2 FCS_RNG RANDOM NUMBER GENERATION 40
8 SECURITY REQUIREMENTS 41
8.1 SECURITY FUNCTIONAL REQUIREMENTS 41
8.1.1 Security attributes 41
8.1.1.1 SCD/SVD Management 42
8.1.1.2 SCD Operational 42
8.1.1.3 IAS ECC Management 42
8.1.1.4 Key Management 42
8.1.2 SFRs drawn for PP 42
8.1.2.1 Phase 6&7 42
8.1.2.2 Phase 7 51
8.1.3 Additional SFRs 53
8.1.3.1 Phase 6 53
8.1.3.2 Phase 7 54
8.1.3.3 Phase 6 & 7 59
8.2 SECURITY ASSURANCE REQUIREMENTS 65
8.2.1 AVA_VAN.5 augmentation 66
8.2.2 ALC_DVS.2 augmentation 66
8.3 SECURITY REQUIREMENTS RATIONALE 67
8.3.1 Security requirement coverage 67
8.3.2 TOE security requirements sufficiency 70
8.3.3 Satisfaction of dependencies of security requirements 74
8.3.3.1 Dependencies 74
8.3.3.2 Justifications for non satisfaction of dependencies 77
9 TOE SUMMARY SPECIFICATIONS 78
9.1 DESCRIPTION 78
9.1.1 SF.RAD_MGT 78
9.1.2 SF.SIG 78
9.1.3 SF.DEV_AUTH 79
9.1.4 SF.ADM_AUTH 79
9.1.5 SF.SM 79
9.1.6 SF.KEY_MGT 80
9.1.7 SF.CONF 80
9.1.8 SF.ESERVICE 81
9.1.9 SF.SAFESTATE_MGT 81

9.1.10 SF.PHYS 81
10 ANNEX A – COMPOSITION WITH THE UNDERLYING JAVACARD PLATFORM 82
10.1 EVALUATION ASSURANCE LEVEL 82
10.2 COVERAGE OF THE ASSUMPTIONS OF THE JAVACARD OPEN PLATFORM (A.PLT VS TOE) 82
10.3 COVERAGE OF THE OSP OF THE JAVACARD OPEN PLATFORM (OSP.PLT VS TOE) 82
10.4 COVERAGE OF THE SECURITY OBJECTIVE OF THE JAVACARD OPEN PLATFORM ENVIRONMENT (OE.PLT VS TOE) 82
10.5 SUPPORT OF THE TOE TSFS BY THE JAVACARD OPEN PLATFORM TSFS (TSF.TOE VS TSF.SFR) 82
10.6 SUPPORT OF THE TOE SFRS BY THE JAVACARD OPEN PLATFORM SFRS (SFR.TOE VS SFR.PLT) 83
10.7 COVERAGE OF THE COMPOSITE ST THREATS BY THE PLATFORM THREATS 86

1 Definitions

ADF	Application Dedicated File
AES	Advanced Encryption Standard
AID	Application Identifier
AMB	Access Mode Byte
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
CSE	Current Security Environment
DAP	Data Authentication Pattern (enable to ensure integrity & authenticity of javacard package when loaded)
CSP	Certificate Service Provider
DAPP	Device Authentication with Privacy Protection
DES	Data Encryption Standard
DF	Dedicated File
DH	Diffie Hellman
DTBS	Data to be signed (Sent by the SCA)
DTBS Representation	Representation of the Data to be signed
EAL	Evaluation Assurance Level
EF	Elementary File
EEPROM	Electrically Erasable Programmable Read Only Memory
FID	File identifier
GP	Global Platform
HI	Human Interface (used to enter the RAD and VAD by the user)
IC	Integrated Chip
ICC	Integrated Chip card
IFD	Interface Device
MAC	Message Authentication code
PIN	Personal Identification Number
PKI	Public Key Infrastructure
RAD	Reference Authentication Data (PIN stored)
RCA	Root Certification Authority
ROM	Read Only Memory
RSA	Rivest Shamir Adleman
RSA CRT	Rivest Shamir Adleman – Chinese Remainder Theorem
SCA	Signature creation Application (Application requiring a qualified signature to the chip)
SCB	Security Condition Byte
SCD	Signature Creation Data (Signature key)
SCP	Secure Channel Procotol
SDO	Security Data Object
SE	Security Environment
SHA	Secure hashing Algorithm
SSCD	Secure Signature Creation Device
SSE	Static Security Environment
SSESP	Static Security Environment for Security Policies
SSO	Single Sign On
SVD	Signature Verification Data (Signature Verification key)
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

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

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

[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)

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

[CC31-2] [1] "Common Criteria for information Technology Security Evaluation, Part 2: Security Functional component", April 2017, Version 3.1 revision 5.

[CC31-3] [1] "Common Criteria for information Technology Security Evaluation, Part 3: Security Assurance components", April 2017, Version 3.1 revision 5.

[FIPS180-3] "FIPS PUB 180-3, Secure Hash Standard"
October 2008 , National Institute of Standards and Technology

[GP2.2.1] Global Platform, Card Specification - Version 2.2.1 – January 2011.

[IASECC] European Card for e-Services and national e-ID Applications - IAS ECC v1.0.1

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

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

[Minidriver] Windows Smart Card Minidriver Specification - Version 7.06 - July 1, 2009

[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] Javacard Open platform certified under reference [PTF_CERTIF]

[PTF_CERTIF] ANSSI-CC-2020/26-R1

[PP0084] Security IC Platform Protection Profile with augmentation packages - Version 1.0 -
BSI-CC-PP-0084-2014

[TR03111] Technical Guideline TR-03111 - Elliptic Curve Cryptography - Version 2.0

[RGS_B1] Référentiel général de sécurité, version 2.0 du 13/06/14 - Annexe B1 - Mécanismes cryptographiques

[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

[SP800-38B] NIST Special Publication 800-38B, Recommendation for Block, Cipher Modes of Operation: The CMAC Mode for Authentication, Morris Dworkin, May 2005

[14890] CEN/EN14890:2013
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

[9797-1] ISO/IEC 9797-1:2011, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher

[11568-2] ISO 11568-2:2012,Financial services - Key management (retail) - Part 2 : symmetric ciphers, their key management and life cycle

3 Security Target Introduction

3.1 Public Security Target Reference

Title	IAS ECC v2, version 1.3, in configuration #2 on ID-One Cosmo v8.2 open platform on NXP P6022M VB - Public Security Target
Reference and version	FQR 110 9185 Ed**5**
Author	IDEMIA
CC version	3.1 revision 5
EAL	EAL5 augmented with AVA_VAN.5 and ALC_DVS.2

3.2 TOE Reference

TOE name	IAS ECC v2, version 1.3, in configuration #2 on ID-One Cosmo v8.2 open platform on NXP P6022M VB
TOE version number	R1.3
Developer name	IDEMIA

Guidance document for preparation	FQR 110 8968 – Clytemnestre-R – AGD_PRE
Guidance document for operational use	FQR 110 8969 - Clytemnestre-R - AGD_OPE
Guidance document for preparation of Platform	FQR 110 8875 - ID-One Cosmo V8.2 - Pre-Perso Guide
Guidance document for operational use of Platform	FQR 110 8885 - ID-One Cosmo V8.2 - Reference Guide FQR 110 8001- ID-One Cosmo V8.1 - Application Loading Protection Guidance FQR 110 8963 - ID-One Cosmo V8.2 - Security Recommendations
Guidance document with Recommendations for compatibility with the "French Qualification Renforcée" framework	FQR 110 9078 - Recommandations pour la compatibilité avec le referentiel de qualification renforcée

Name of [PLT]	Plateforme JavaCard de la carte à puce *ID-One Cosmo V8.2* sur composant P6022y VB (NXP P60D145)
Certificate	[PTF_CERTIF]

The TOE identification (AID and version) is described in in section 6.3 of [AGD_PRE].

3.3 TOE overview

3.3.1 TOE Type

The Target of Evaluation is a smartcard which is configured as a Secure signature creation Device (SSCD), used to create advanced or qualified signature in the sense of EC/1999/93.

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 of four javacard components:

- a javacard Applet ([Applet]);
- a javacard API ([API]);
- two javacard Interfaces ([Interface]);

[Applet] relies on

- [API] which provides a wide range of services enabling to manage the files and cryptographic objects;
- [Interface] which provides the mechanisms for data sharing with other applets;
- Javacard API provided by the underlying javacard open platform;

3.3.2 Logical scope

The TOE is made up of:

- The underlying javacard open platform
- The javacard code ([Applet], [API] and [Interface])

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 complies with [AN10] and the specific requirements of the TOE stated in the guidance documents.

The logical scope of the TOE may be depicted as follows:

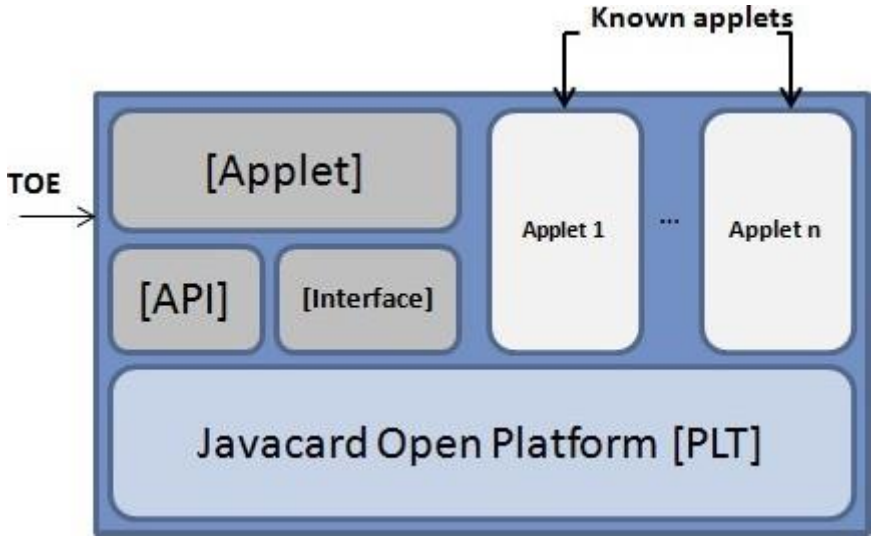


Figure 1 - Limits of the TOE

3.3.3 Physical scope

The TOE is physically made up of several components:

- the javacard open platform **[PLT],** which contains in its ROM code the javacard packages **[Applet]**, **[API]** and **[Interface]**;
- A potential patch **[patch]** loaded in EEPROM. If a functional patch is required, its reference will be included in a maintenance report;
- the other applets that may potentially be loaded on the javacard open platform **[PLT]** at any time;

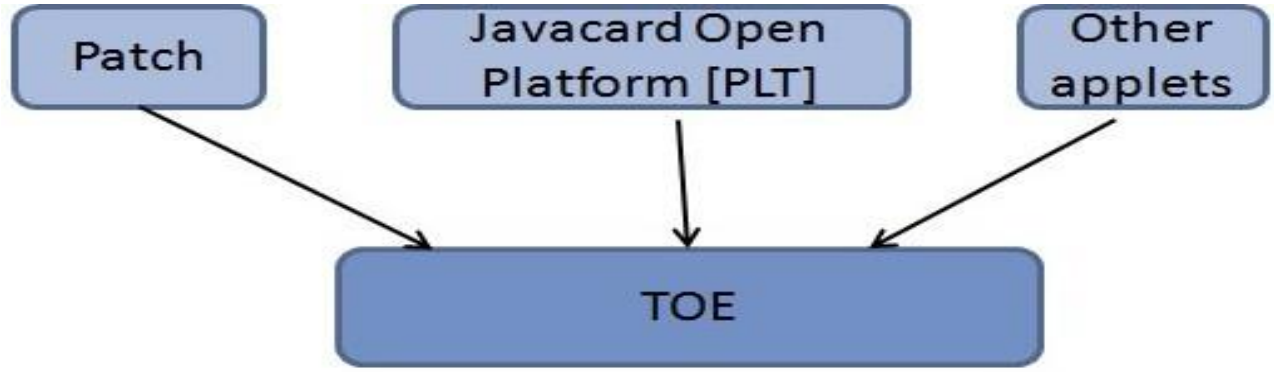


Figure 2 - Physical scope of the TOE

The patch, if present, is self protected (encrypted and signed). The other applets must fulfill the requirements stated in [AN10] and in the guidance documentation of the TOE.

Once constructed, the TOE is a bare microchip with its external interfaces for communication. The physical medium on which the microchip is mounted is not part of the target of evaluation because it does not alter nor modify any security functions of the TOE.

The TOE may be used on several physical medium

- within an inlay, or eCover;
- in a plastic card;
- within a USB key;
- ….;

3.3.4 Required non-TOE hardware/software/firmware

The TOE is a Secure 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 card reader.

3.3.5 Usage and major security features

The TOE intended usage is to be used as a “secure signature creation device” with key generation and/or key import, with respect to the European directive EC/1999/93.

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

- perform basic, advanced and qualified signature;
- authenticate the cardholder based on a PIN and/or Biometric data verification;
- authenticate one (or several) administrator(s) of the TOE, that may have special rights to administrate the SCD and SVD (generation, import), using either symmetric and/or asymmetric mechanisms, or PIN and/or Biometric data verification;

- establish trusted channel, protected in integrity and confidentiality, with Trusted IT entities such as a SCA or a CSP. It may be realized by means of symmetric and/or asymmetric mechanisms;

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

- A super Administrator (TOE_Administrator) has special rights to administrate the signature creation function, the mode of communication, and the type of cryptographic mechanisms to use.
- SCD/SVD pairs and other cryptographic objects may be generated and/or imported after issuance at any time, and in particular, they may be updated during the TOE life cycle.
- The TOE may be used to realize digital signature in contact and/or contactless mode.
- eServices features are added, enabling the cardholder to perform C/S authentication, Encryption key decipherment….
- A complete access control over objects is ensured, whatever their type is : File or cryptographic objects (PIN, keys,…), ensuring it is not possible to bypass the access rules.

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

- Electronic 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);

3.3.6 Scope of evaluation

The scope of evaluation covers the following features:

- Features covered by [SSCD2]], [SSCD3]
- Authentication mechanisms based on cryptographic scheme
- Unblocking of RAD
- Management of the other keys (authentication and e-services)

3.4 TOE Description

The TOE is compliant with the specification [IASECC], and is enhanced with the following features:

- The TOE supports user authentication based on Biometric comparison. Two modes of operations, are possible: either a 1:1 Biometric comparison, or a 1:n comparison can be made. These modes of operations are compliant to [14890] and [7816-4]
- The TOE supports Elliptic curves cryptography for electronic signature, encryption key decipherment, and C/S authentication. These modes of operations are compliant to [14890].
- The TOE supports several modes of operation for the data hashing. The data may also be fully hashed on card or off card. These modes of operations are compliant to [14890].
- The TOE supports secure messaging and authentication scheme based on AES block Cipher. These modes of operations are compliant to [14890].
- The TOE supports several features required by [Minidriver]

3.4.1 Data structure

The TOE manages two types of structures:

- The Files, compliant with [7816-4]
- The Security Data Objects, which are secure containers storing cryptographic data (PINs, Keys,…)

3.4.1.1 File and File System

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

- Transparent File - EF
- Application Dedicated File - ADF

- Dedicated File - DF

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 Root file (or Master File), it is the default selected file at reset. Under the Root file, are located the Application Dedicated File.

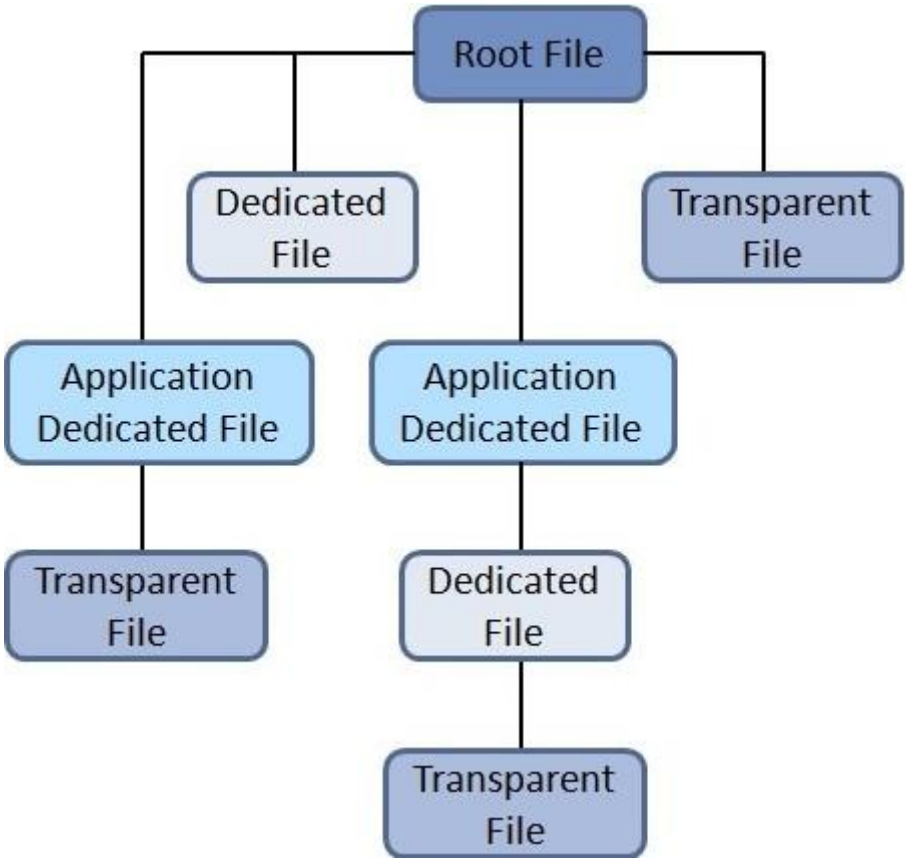


Figure 3 - Exemple of File System structure

The Root, as well as each ADF and DF, may contain up Elementary File (EF) or Security Data Object (SDO). Each of them may contain up to 255 files (EF or DF) and 31 SDO of each type.

The TOE allows to

- create, delete, activate, deactivate, and terminate any type of file (except the Application dedicated file), which update the File System.
- read, update, resize any transparent file (EF)
- move within the File Structure by use of file selection

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

- Access conditions
- File identifier
- Location within the File System
- Size (for EF)

The management of the file system is fully described in [IASECC].

3.4.1.2 Security Environment

The TOE handles Security Environments. Three types of Security Environment may be sorted out:

- Static Security Environment - SSE
- Static Security Environment for Security Policies – SSESP
- Current Security Environment - CSE

Basically a security environment contains several couple of cryptographic data, each of them containing:

- One or several key identifier : KEY_ID
- an algorithm identifier : ALGO_ID
- a mode of usage : USE

These cryptographic data may be used to:

- load a pre-defined cryptographic context to perform a cryptographic operation (for signature, for C/S authentication,…). It is the case of a SSE.
- define an access condition to fulfill before granting an access right: the key defined by the identifier KEY_ID shall be used with the algorithm ALGO_ID and with the mode USE to grant an access right. It is the case of a SSESP.
- Store the current cryptographic context required to realize a given service. It is the case of the CSE.

The SSESP and SSE are bound to an ADF and are stored in security Data Objects located within an Application dedicated file (ADF). The CSE is unique for the TOE at any moment

3.4.1.3 Security data Objects

The TOE handles as well cryptographic data objects, called Security Data Objects (SDO), dedicated to store the keys, the PIN, the Biometric template, the Diffie Hellmann parameters and the Security Environments, as well as their attributes. The following types of SDO are available:

- SDO PIN contains a Personal identification Number
- SDO BIO contains one or several Biometric template
- SDO RSA Public Key contains a RSA Public Key
- SDO RSA Private Key contains a RSA Private Key
- SDO ECC Public Key contains an ECC Public Key
- SDO ECC Private Key contains an ECC Private Key
- SDO Security Environment contains a Security Environment
- SDO Symmetric DES Key Set contains a Symmetric DES Key Set
- SDO Symmetric AES Key Set contains a Symmetric AES Key Set
- SDO Diffie Hellmann parameters contains a set of Diffie Helmann Domain parameters

The SDO may be located in any dedicated file (DF) or Application Dedicated file (ADF).

The TOE enables to create, update and use any of these SDO. The way the SDO may be used depends on its type:

- SDO PIN and SDO BIO may be changed, reset, verified
- SDO RSA Public Key may be used to verify a certificate
- SDO RSA Private Key and SDO ECC Private key may be used to sign, perform a C/S authentication or decrypt a cryptogram
- SDO Security Environment may be changed, reset, verified
- SDO Symmetric DES Key Set and SDO Symmetric AES Key Set may be used to verify an external authentication or to perform a mutual authentication and establish a trusted channel
- SDO Diffie Hellmann parameters may be used to establish a secure channel (without authentication)

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

- Access conditions
- Location within the File System
- Size
- Type
- Secret value

- Usage counter and tries counter
- Algorithm to be used

The management of SDO is fully described in [IASECC].

3.4.2 Access Control Management

One of the Core features of the TOE is to provide access control management on any operations on any objects it handles (Files of SDO).

The Access conditions encoding is the compact encoding described in [7816-4], enhanced as described in [IASECC]. It relies on access rules encoded by means on Access Mode Bytes (AMB) and Security Conditions Bytes (SCB) as described in [7816-4] and [IASECC].

Prior to granting access to a given operation, the TOE checks the requested access rights are fulfilled. Basically, an Access condition is granted if the security conditions are fulfilled. An access condition is a combination of security conditions based on identified keys/PIN/BIO/secrets:

- User Authentication (by PIN or Biometric comparison). It is used to authenticate the cardholder or an external entity administrator
- Authentication of an external entity administrator
- Mutual authentication with a trusted IT entity
- Communication protected in integrity and confidentiality

3.4.3 Authentication of entities

The TOE allows the authentication of several entities in order to grant them some rights.

- User Authentication (by PIN or Biometric comparison). It is used to authenticate the cardholder or an external entity administrator
- Authentication of an external entity administrator (based on symmetric or asymmetric scheme)
- Mutual authentication with an external entity and establishment of a trusted channel protected in integrity and confidentiality (based on symmetric or asymmetric scheme)
- Personalization Agent authentication (for the phase 6)
- TOE Administrator authentication (in phase 7)

These authentication mechanisms are the cornerstone for the access control mechanisms used to grant access to resources (Files or SDO).

3.4.4 Electronic Services

The TOE supports as well several electronic services:

- C/S authentication: this feature enables to authenticate the TOE to an external entity.
- Digital signature: this feature enables the cardholder to electronically signs documents. The signature may be either advanced or qualified (compliant with [SSCD2] and [SSCD3]).
- Encryption key decipherment: this feature enables the cardholder 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 cardholder's computer sends the encrypted encryption key to the TOE to get the plain encryption key.

3.4.5 Administration of the TOE

The TOE offers administration services. Upon successful authentication, the TOE Administrator may modify the following attributes:

- Communication medium: the administrator may restrict the ability to communicate with the TOE in contact and/or contactless mode.
- Hashing method to be used for digital signature: the administrator may restrict the ability to perform electronic signature (advanced or qualified) on DTBS-representation partly computed by the TOE. In such case, the digital signature will only be done with last round of data hashing done on the TOE.

- Authentication mechanism to be used: the administrator may restrict the cryptographic means to be used by the TOE to authenticate external entities (Administrator or IT entity): either symmetric and/or asymmetric cryptography.
- Identification of the TOE : the administrator is entitled to identify the TOE
- Biometric threshold : the administrator can modify the biometric threshold

3.4.6 Single Sign on feature (SSO)

The TOE may also behave as a Single Sign on (SSO). It provides access points to any other applet willing to use authentication services based on a PIN stored in the Root File (or Master File). In particular it is possible to:

- Check a PIN
- Change a PIN
- Reset a PIN
- Retrieve the remaining tries counter
- Retrieve the validation status

This feature is used for instance when the PIN(s) is shared with a legacy application. Even though the TOE offers these entry points, it does still enforce access control in the same way it does when it receives incoming APDU to use a PIN.

3.5 Life Cycle

With respect to the Life cycle envisioned in [PP0084], seven different phases may be sorted out. The life cycle of the composite TOE may be depicted as follows:

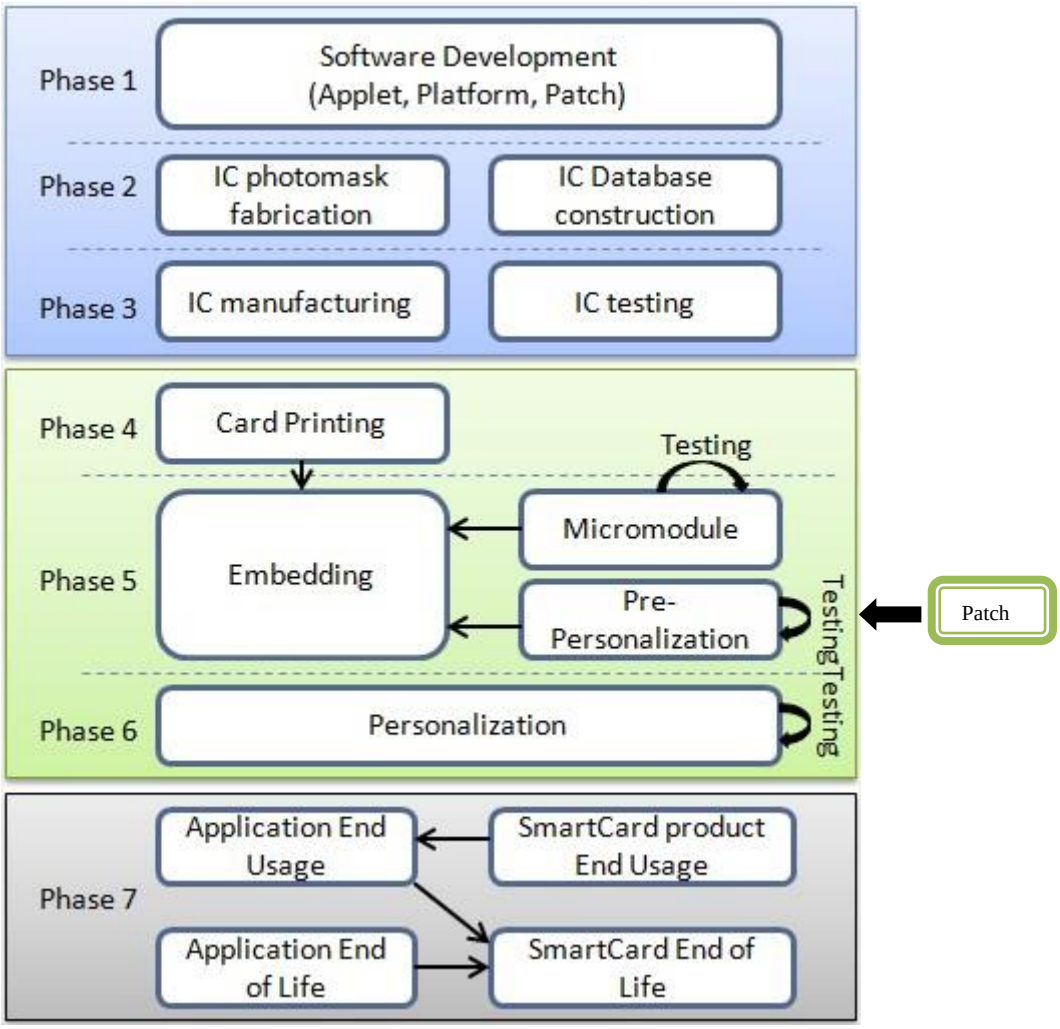


Figure 4 - TOE life cycle

The point of delivery of the TOE is the end of phase 3. At this moment, the TOE is self protected, but not constructed.
The point of delivery of the personalization key required to authenticate with the TOE in phase 6, is the end of phase 5.

The TOE Life cycle may be splitted in three steps

- Development (phase 1 to 3);
- Production (phase 4 and 5);
- Operational state (phase 6 and 7);

3.5.1 Development

The development of the TOE takes place in phase 1 to 3. In this step, the parts of TOE are designed, tested and manufactured. This step is covered by [ALC] tasks.

TOE development sites:

- IC development : covered by IC certification
- Platform Code: Courbevoie, Pessac
- Application Code: Courbevoie

3.5.1.1 Software development (phase 1)

This development environment of the Javacard Applet, the patch if any and javacard open platform (JOP) is enforced by IDEMIA.

The confidentiality and integrity of the cap files, the patch and of the javacard open platform is covered by the evaluation of the development premises of IDEMIA.

To ensure security, access to development tools and products elements (PC, card reader, documentation, source code...) is protected. The protection is based on measures for prevention and detection of unauthorized access. Two levels of protection are applied:

- Access control to IDEMIA offices and sensitive areas.
- Access to development data through the use of a secure computer system to design, implement and test software

At the end of this phase, the code of the javacard applet is delivered to the javacard open platform development team, in order to be stored in the ROM code. The software development phase of the javacard open platform is covered by [PLT].

3.5.1.2 Hardware development (Phase 2)

In this phase, the underlying integrated circuit is developped. This phase takes place at the manufacturing site of the silicium provider.

The confidentiality and integrity of the javacard packages and javacard open platform is covered by the evaluation of the development premises of the silicium manufacturer (see [PLT])

3.5.1.3 Javacard open platform manufacturing (phase 3)

In this phase, the code of the javacard open platform (JOP) and the applet are masked on the IC. This phase takes place at the manufacturing site of the silicium provider.

The confidentiality and integrity of the javacard packages and javacard open platform is covered by the evaluation of the development premises of the silicium manufacturer (see [PLT]).

Depending on the choice made for the optional code loading, it may be loaded during this phase.

At the end of phase 3, the javacard open platform (JOP) and the TOE are self protected: all its security functions are activated. The point of delivery of the TOE is the end of phase 3.

3.5.2 Production

The production environment encompasses the preparation of the TOE and the management of the personalization key used to personalize it.

During this step, the following operations are made:

- The chip is mounted on a physical layout (card, USB token...)
- The javacard open platform is prepersonalized
- The javacard open platform is personalized
- The personalization key is loaded on the TOE
- The applet is instantiated

This step is covered by [AGD_PRE] tasks for the TOE, and by [ALC] for the management of the personalization key in its environment.

3.5.2.1 Packaging and initialization (phase 4)

This phase is performed by the Manufacturing Agent, which controls the TOE that is in charge of the packaging and initialization of the Javacard open platform (JOP).

This phase spans the phase 4 of the Javacard open platform (JOP) life cycle and is covered by [AGD_PRE] tasks of [PLT].
All along this phase, the TOE is self-protected as it requires the authentication of the Manufacturing Agent prior to any operation.

3.5.2.2 Preparation (phase 5)

This phase is performed by the Manufacturing Agent, which controls the TOE, in the all IDEMIA manufacturing sites (as for example, but not limited to: Vitré (France – 35), Shenzhen (China), Haarlem (Netherlands), Noida (India) and Ostrava (Czech republic)). The procedures and the IT infrastructure ensure the integrity and authenticity of the keys used to get authenticated with the TOE.

This phase spans the following phases of the javacard open platform (JOP):

- Phase 5
- Phase 6
- Phase 7

The following process is applied during this phase

- a non-security patch [patch] (patch code that has no impacts on product auto-protection) is loaded in the javacard open platform (JOP) (if needed);
- the javacard open platform (JOP) is switched in phase 5 and the applet may be instantiated in this phase;
- the javacard open platform (JOP) is switched in phase 6 and the applet may be instantiated in this phase;
- the javacard open platform (JOP) is switched in phase 7 and the applet may be instantiated in this phase;

Before the patch is loaded in the javacard open platform, the TOE is made of two elements (the patch and the javacard open platform). Once it is loaded, the TOE is the single javacard open platform containing the patch. Moreover, during this phase, any other applet may be loaded at any time (phase 5, 6 or 7 of the javacard open platform), provided they fulfill the requirements laid down in [AN10]. At the end of this phase, the javacard open platform is switched in phase 7 (DAP enforced)

All along this phase, the TOE is self-protected as it requires the authentication of the Manufacturing Agent prior to any operation.

3.5.3 Operational state

3.5.3.1 Applet pre-personalization (phase 6)

This phase is performed by the Personalization Agent, which controls the TOE. During this phase, the javacard applet is prepared as required by P.TOE_Construction.

All along this phase, the TOE is self-protected as it requires the authentication of the Personalization Agent prior to any operation.

3.5.3.2 TOE personalization (phase 6)

This phase is performed by the Personalization Agent, which controls the TOE, which is in charge of the javacard applet personalization.

All along this phase, the TOE is self-protected as it requires the authentication of the Personalization Agent prior to any operation.

This phase may not necessarily take place in a manufacturing site, but may be performed anywhere. The Personalization Agent is responsible of ensuring a sufficient level of security during this phase.

The javacard applet is personalized according to [AGD_PRE], and the following operations are made: creation of applicative data (SCD, SVD, RAD, File,...) and the TOE_Administrator Agent key is loaded.

At the end of phase 6, the TOE is constructed.

3.5.3.3 TOE Usage (phase 7)

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

During this phase, the TOE may be used to create a secure signature and manage the SCD, the SVD and the RAD.

3.5.4 Coverage of the different Life cycle state by the assurance components [AGD] & [ALC]

The following phases of the life cycle are covered as follows:

<table>
<tr><th>Steps</th><th>Life cycle State</th><th>TOE : covered by</th><th>Personalisation key : covered by</th></tr>
<tr><td rowspan="4">Development</td><td>Phase 1</td><td>ALC [PLT]
ALC [Applet]</td><td>N/A</td></tr>
<tr><td colspan="3">Patch is self protected</td></tr>
<tr><td>Phase 2</td><td>ALC [PLT]
ALC [Applet]</td><td>N/A</td></tr>
<tr><td>Phase 3</td><td>ALC [PLT]
ALC [Applet]</td><td>N/A</td></tr>
<tr><td colspan="4">Patch is loaded
TOE is self protected</td></tr>
<tr><td colspan="4">Point of delivery of the TOE</td></tr>
<tr><td rowspan="2">Production</td><td>Phase 4</td><td>AGD_PRE [PLT]</td><td>N/A</td></tr>
<tr><td>Phase 5</td><td>AGD_PRE [PLT]
AGD_OPE [PLT]</td><td>ALC [Applet]</td></tr>
<tr><td colspan="4">Point of delivery of the personalization key</td></tr>
<tr><td colspan="4">Patch is loaded</td></tr>
<tr><td rowspan="3">Operational</td><td>Phase 6</td><td>AGD_OPE [PLT]
AGD_PRE [Applet]</td><td>N/A</td></tr>
<tr><td colspan="3">TOE is constructed</td></tr>
<tr><td>Phase 6</td><td>AGD_OPE [PLT]
AGD_PRE [Applet]</td><td>N/A</td></tr>
</table>

	Phase 7	**AGD_OPE [PLT]** **AGD_ OPE [Applet]**	**N/A**

The point of delivery of the TOE is the end of phase 3, and the point of delivery of the personalization key is the end of phase 5. The security of the patch loading (done after phase 3) is fully enforced by technical security measures that have been evaluated in [PLT]. Therefore, phase 4 to 6 are fully covered by [AGD_PRE] and [AGD_OPE], except the personalization key management in the environment which is covered by [ALC].

3.5.5 State of the TOE depending on the phase

Life cycle State	**TOE**		**Personalisation key**	
	Self protected	**constructed**	**stored in**	**Protected by**
Phase 1	**No**	**No**	**N/A**	**N/A**
Phase 2	**No**	**No**	**N/A**	**N/A**
Phase 3	**No**	**No**	**N/A**	**N/A**
Phase 4	**Yes**	**No**	**N/A**	**N/A**
Phase 5	**Yes**	**No**	**Manufacturing centre**	**ALC[Applet]**
Phase 6	**Yes**	**Yes**	**N/A**	**N/A**
Phase 7	**Yes**	**Yes**	**N/A**	**N/A**

3.5.6 Mapping with the Users

For each of these phases, the following subjects may interact with the TOE

Life cycle phase	**Subject interacting with the TOE**
Phase 1	IDEMIA
Patch ,if it exists, is self protected	
Phase 2	IDEMIA
Phase 3	IDEMIA
TOE is self protected	
Phase 4	Manufacturing Agent Offcard
Phase 5	Manufacturing Agent Offcard
Phase 6	Personalization Agent Offcard
TOE is constructed	
Phase 6	Personalization Agent Offcard
Phase 7	Users

4 Conformance Claim

4.1 CC and package Conformance claim

This 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 to the extended part. ▪ FCS.RNG.1: "Random number generation" ▪ FPT_EMS.1: "TOE Emanation"
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.

This security target also addresses the manufacturing and personalization phases at TOE level (cf. TOE life cycle presented in §3.5. The 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 Life cycle conformance

The life cycle of the TOE is described in §3.5. This chapter demonstrates the mapping of the TOE's life cycle with the one described in the protection profiles.

Life cycle phase of the TOE	Life cycle phase with respect to the protection profiles
Phase 1	Development phase: SSCD Development
Patch is self protected	
Phase 2	Development phase: SSCD Production
Phase 3	Development phase: SSCD Production
TOE is self protected	
Phase 4	N/A
Phase 5	N/A
Patch is loaded on the Javacard open platform **TOE is self protected**	
Phase 6	Usage phase: SSCD Preparation
TOE is constructed	
Phase 6	Usage phase: SSCD Preparation
Phase 7	Usage phase: SSCD Operational use

4.3.2 Additional assets

All assets from the protection profiles are included in this security target. The following assets have been added:

Keys:

1. Private or secret keys used to authenticate an external user or entity, or to perform eServices. Their integrity and confidentiality must be maintained.

2. public key used to perform eServices. Their integrity must be maintained.

4.3.3 Additional Roles

The roles from protection profiles are maintained in this security target; however the following refinements for the role R.Admin have been added:

- Personalisation Agent
- User_Admin
- SCA
- HID
- IFD

4.3.4 Additional threats

All the threats from the protection profiles are maintained in this security target. The following policies have been added:

- T.Key_Divulg Storing, copying, and releasing of a key stored in the TOE
- T.Key_Derive Derive a key
- T.TOE_PublicAuthKey_Forgery Forgery of the public key of a TOE authentication key
- T.Authentication_Replay Replay of an authentication of an external entity

4.3.5 Additional OSPs

All the Policies from the protection profiles are maintained in this security target. The following policies have been added:

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

- P.TOE_PublicAuthKey_Cert Certificate for asymmetric TOE authentication keys
- P.TOE_Construction Construction of the TOE by the Personalization Agent
- P.eServices Provision of eServices

4.3.6 Additional objectives

4.3.6.1 Additional Security objectives for the TOE

All the security objectives for the TOE from the protection profiles are maintained in this security target. The following objectives have been added:

- OT.Authentication_Secure Secure authentication mechanisms
- OT.SCD/SVD_Management Management of SCD/SVD
- OT.Key_Lifecycle_Security Life cycle security of the keys stored in the TOE
- OT.Keys_Secrecy Secrecy of Keys
- OT.TOE_AuthKey_Unique Uniqueness of the TOE authentication key(s)
- OT.Lifecycle_Management Management of the life cycle
- OT.eServices Provision of eService

4.3.6.2 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. The following objectives have been added:

- OE.LinkSCD_QualifiedCertificate Link betwaan SCD stored in the TOE and the relevant qualified certificate
- OE.AuthKey_Transfer Secure transfer of authentication key(s) to the TOE
- OE.AuthKey_Unique Uniqueness of the authentication key(s)
- OE.TOE_PublicKeyAuth_Transfer Secure transfer of public authentication key(s) of the TOE
- OE_TOE_Construction Construction of the TOE by the Personalisation_Agent

4.3.7 Additional SFRs

All the SFRs from the protection profiles are maintained. The following SFRs have been added to cover supplemental features:

Additional SFRs	**Rationale**
FCS_CKM.1 /Session keys	Generation of secure messaging session keys
FCS_CKM.1/Keys	Generation of authentication and eServices keys
FCS_CKM.4/Session keys	Destruction of secure messaging session keys
FCS_COP.1/DH Computation	Cryptographic operation : Diffie Hellman
FCS_COP.1/SM in Confidentiality	Cryptographic operation : protection in confidentiality of APDU
FCS_COP.1/SM in Integrity	Cryptographic operation : protection in integrity and authenticity of APDU
FCS_COP.1/data hashing	Cryptographic operation : Data hashing
FCS_COP.1/C/S Auth	Cryptographic operation : C/S Authentication
FCS_COP.1/Enc key decipherment	Cryptographic operation : Encryption key decipherment
FCS_COP.1/Sym Role Auth	Cryptographic operation : symmetric role authentication
FCS_COP.1/Sym Device Auth	Cryptographic operation : symmetric device authentication
FCS_COP.1/Certificate Verification	Cryptographic operation : Certificate verification
FCS_COP.1/Asym Role Auth	Cryptographic operation : asymmetric role authentication
FCS_COP.1/Asym Internal DAPP Auth	Cryptographic operation : asymmetric internal DAPP Authentication
FCS_COP.1/Asym External DAPP Auth	Cryptographic operation : asymmetric external DAPP Authentication
FCS_COP.1/GP Auth	Cryptographic operation : GP authentication
FCS_COP.1/GP secret data protection	Cryptographic operation : GP secret data protection
FCS_RNG.1	Cryptographic operation : Random number generation
FDP_ACC.1/IASECC Administration	Access control policy for the administration operation of IAS ECC
FDP_ACC.1/Key Management	Access control policy for the key management operations
FDP_ACF.1/IASECC Administration	Access control rules for the administration operation of IAS ECC

Additional SFRs	Rationale
FDP_ACF.1/Key Management	Access control rules for the key management operations
FDP_ETC.1/Keys	Export of keys
FDP_ITC.1/ Keys	Import of keys
FIA AFL.1/Auth keys	Management of wrong authentication with mechanisms based on cryptographic keys
FMT_MSA.1/TOE Management	Management of Access rights for IAS ECC administration operations
FMT_MSA.1/Key Management	Management of Access rights for key management operations
FMT_MTD.1/SCD and SCD_ID	Link between a SCD and an identifier
FMT_MTD.1/TOE Serial number	Loading of the TOE serial number
FMT_MTD.1/TOE State	Transition of the life cycle of the TOE from phase 6 to phase 7
FMT_MTD.1/Unblock	Unlocking of RAD by the administrator

4.3.8 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: User Data

1. **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.

2. **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.

3. **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 : TSF Data

1. **Keys**:

    a. Private or secret keys used to authenticate an external user or entity, or to perform eServices. Their integrity and confidentiality must be maintained

    b. public key used to perform eServices. Their integrity must be maintained.

Note: Diffie Hellman parameters are considered as keys in the rest of the document.

2. **RAD:** Reference PIN code or biometrics authentication reference used to identify and authenticate the End User (integrity and confidentiality of RAD must be maintained)

3. **VAD:** PIN code or biometrics data entered by the End User to perform a signature operation (confidentiality and authenticity of the VAD as needed by the authentication method employed)

4. **Session keys:** Keys computed for secure messaging and used to ensure confidentiality and integrity of data.

5.1.2 Subjects

1. 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.

2. 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. The CSP (Certificate Service Provider) who is in charge of generating SCD/SVD key pair and importing SCD also counts as Administrator. (Subject from PP).

   The following refinements of R.Admin may appear in this document:
   - Personalisation Agent: Administrator in charge of the personalisation in phase 6
   - User_Admin: User with administrative rights in phase 7
   - SCA: Signature Creation application
   - HID: Human Interface Device
   - IFD: Interface Device

3. TOE_Administrator: Administrator in phase 7 in charge of the TOE management (Additional Subject).

4. Signatory: User who holds the TOE and uses it on their 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. (Subject from PP).

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, copy an authentication or eService key stored in the TOE outside the TOE. An authentication key may be either used to authenticate an external entity or the TOE, and may be symmetric or asymmetric. An attacker can release an authentication or eService key during generation, storage and use in the TOE.

5.2.2.2 T.Key_Derive *Derive a key*

An attacker derives an authentication key (of the TOE or an external entity) or eService key 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.

5.3 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.

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

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 (**Personalisation Agent**, **R.Admin**, **R.Sigy**), or the trusted IT entity involved in the updating process (CSP**)** shall ensure an unambiguous link between the (qualified) certificate(s) and the matching SCD(s). This link might be figured out by a PKCS#15 structures, an XML structure, an identifier .linking the file containing the (qualified) certificate or the URL hosting them to the SCD(s) loaded in the TOE. In particular, it implies this link 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 its private key used for authentication is genuine.

5.3.2.3 P.TOE_Construction *Construction of the TOE by the Personalization Agent*

The recommendations indicated in [**AGD_PRE**] required to construct the TOE are correctly applied.

5.3.2.4 P.eServices Provision of eServices

The TOE provides eServices Mechanisms enabling to:

- decrypt encryption keys
- authenticate the TOE
- verify CVC certificates

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

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 authorised 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 initialisation, personalisation 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.

6.1.2 Additional Security Objectives for the TOE

6.1.2.1 OT.Authentication_Secure *Secure authentication mechanisms*

The TOE provides strong mechanism to authenticate external users/entity and mechanisms to establish a strong trusted channel with an external IT entity. The authentication protocols rely on cryptographic schemes that are based on either symmetric or asymmetric cryptography. The TOE uses freshly generated random number in the authentication mechanism in order to avoid replay attacks. The authentication protocols ensure that the cryptogram can not be forged without the knowledge of the authentication key, and that they can not be reconstructed from the authentication cryptograms. The trusted channel ensures integrity, authenticity, and confidentiality of the data using strong encryption techniques. The trusted channel 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.

6.1.2.2 OT.SCD/SVD_Management *Management of SCD/SVD*

The TOE enables to manage SCD/SVD. Each key (pair) and RAD may be created at any time and used to perform qualified signature during the TOE life time. Several SCD, SVD, and RAD may be present on the TOE and used by the same holder. The TOE guarantees the SCD, SVD and RAD are independent from each other.

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

The TOE shall detect flaws during the initialisation, personalisation and operational usage. The TOE shall provide safe destruction techniques for the authentication keys (of the TOE and/or the external entities) and eServices keys it stores in case of erasure, re-import or re-generation.

6.1.2.4 OT.Keys_Secrecy *Secrecy of Keys*

The secrecy of the authentication keys (of the TOE and/or the external entities) and eServices keys stored in the TOE is reasonably assured against attacks with a high attack potential.

6.1.2.5 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.

6.1.2.6 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 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 **RAD** (s) may be created and loaded
- **SVD** and public keys may be exported

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

6.1.2.7 OT.eServices *Provision of eServices*

The TOE provides eServices Mechanisms enabling to:

- decrypt encryption keys
- authenticate the TOE
- verify CVC certificates

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

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.SSCD_Prov_Service *Authentic SSCD provided by SSCD Provisioning Service*

The SSCD Provisioning Service shall initialize and personalize for the signatory an authentic copy of the TOE and deliver this copy as SSCD 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.

6.2.2 Additional security objectives for the operational environment

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

The role in charge of creating and updating the SCD (**Personalisation Agent**, **R.Admin**, **R.Sigy**), or the trusted IT entity involved in the updating process (the **CSP)** shall ensure an unambiguous link between the (qualified) certificate(s) and the matching SCD(s). This link might be figured out by a PKCS#15 structures, an XML structure, an identifier .linking the file containing the (qualified) certificate or the URL hosting them to the SCD(s) loaded 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 by the retrieval of the public key according to certain rules imposed to the TOE holders.

6.2.2.5 OE_TOE_Construction *Construction of the TOE by the Personalisation_Agent*

The Personalization Agent in charge of administrating the TOE in phase 6 shall be a trusted person and shall be skilled enough to correctly apply the recommendations indicated in [**AGD_PRE**]. These recommendations are required to construct the TOE.

6.3 Security Objectives Rationale

6.3.1 Security objectives 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	P.CSP_QCert	P.QSign	P.Sigy_SSCD	P.Sig_Non-Repud	P.LinkSCD_QualifiedCertificate	P.TOE_PublicAuthKey_Cert	P.TOE_Construction	P.eServices	A.CGA	A.SCA	A.CSP
OT.Lifecycle_Security					X							X		X	X							
OT.SCD/SVD_Auth_Gen		X												X								
OT.SCD_Unique							X							X	X							
OT.SCD_SVD_Corresp				X								X			X							
OT.SCD_Auth_Imp	X											X		X								
OT.SCD_Secrecy	X		X											X	X							
OT.Sig_Secure		X					X						X	X	X							
OT.Sigy_SigF					X								X	X	X							
OT.DTBS_Integrity_TOE					X	X								X	X							
OT.EMSEC_Design			X											X	X							
OT_Tamper_ID			X												X							
OT_Tamper_Resistance			X											X	X							
OT.Authentication_Secure									X		X											
OT.SCD/SVD_Management															X							
OT.Key_Lifecycle_Security								X														
OT.Keys_Secrecy			X					X														
OT.TOE_AuthKey_Unique									X													
OT.Lifecycle_Management					X																	
OT.eServices																			X			
OE.SVD_Auth				X											X					X		
OE.CGA_QCert							X					X	X		X					X		
OE.SSCD_Prov_Service														X	X							

	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	P.CSP_QCert	P.QSign	P.Sigy_SSCD	P.Sig_Non-Repud	P.LinkSCD_QualifiedCertificate	P.TOE_PublicAuthKey_Cert	P.TOE_Construction	P.eServices	A.CGA	A.SCA	A.CSP
OE.HID_VAD					X																	
OE.DTBS_Intend					X	X							X		X						X	
OE.DTBS_Protect					X	X									X							
OE.Signatory					X										X							
OE.SCD/SVD_Auth_Gen	X											X		X	X							X
OE.SCD_Secrecy	X													X	X							X
OE.SCD_Unique		X					X							X	X							X
OE.SCD_SVD_Corresp				X								X			X							X
OE.LinkSCD_QualifiedCertificate															X	X						
OE.AuthKey_Transfer								X														
OE.AuthKey_Unique									X													
OE.TOE_PublicKeyAuth_Transfer										X							X					
OE.TOE_Construction																		X				

6.3.2 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

T.SigF_Misuse (*Misuse of the signature creation function of the TOE*) addreses the threat of misuse of the TOE signature creation function to create SDO by others 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, personalisation 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 Personalisation Agent control, it can not 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_PublicAuthKey_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.

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, personalisation and operational usage,
- **OT.SCD_SVD_Corresp**, which requires to ensure the correspondance between the SVD and the SCD during their generation,
- **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,

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, personalisation 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.

OT.SSCD_Prov_Service ensures that the signatory obtains an authentic copy of the TOE, initialised and personalised SSCD from an SSCD-provisioning service.

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.SSCD_Prov_Service ensures that the signatory obtains an authentic copy of the TOE initialized and personalized as an SSCD-provisioning service.
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).
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 Personalisation Agent control, it can not be misused to sign on behalf of the legitimate Signatory.
OE.LinkSCD_QualifiedCertificate and **OT.SCD/SVD_Management** 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.

OT.SCD/SVD_Management ensures that the TOE create signature with the SCD that has been selected by the SCA. As such it ensures the signature is always created with the SCD matching the (qualified) certificate selected by the SCA, avoiding any mismatch between SCD and (qualified) certificate, that may cause the signature to be repudiated.

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_PublicAuthKey_Transfer**.

P.TOE_Construction (*TOE construction*) ensures that all the recommendations indicated in [AGD_PRE] are applied for the construction of the TOE in phase 6. It is addressed by **OE.TOE_Construction**.

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

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

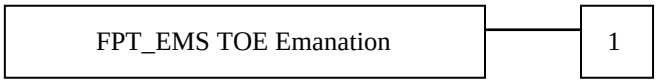
7.1 FPT_EMS TOE Emanation

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*].

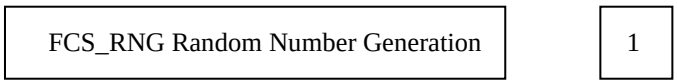
FPT_EMS.1.2 The TSF shall ensure [assignment*: type of users*] are unable to use the following interface [assignment*: type of connection*] to gain access to [assignment*: list of types of TSF data*] and [assignment*: list of types of user data*].

7.2 FCS_RNG Random Number Generation

A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs.

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_ RNG.1 Random Number Generation has two constituents:

- FCS_RNG.1.1 Random number generator type
- FCS_RNG.1.2 Random number quality

Management:

There are no management activities foreseen

Audit:

There are no actions defined to be auditable

FCS_RNG.1 *Random Number Generation*

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

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic hybrid] random number generator that implements: [assignment: list of security capabilities].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: a defined quality metric].

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
S.Admin	IAS ECC Management	Medium	Contact Contactless
		HashOffCard Management	Authorized Not authorized
		SymAuthMechanisms Management	Authorized Not authorized
		AsymAuthMechanisms Management	Authorized Not authorized
S.User	Key Management	Key import Management	Authorized Not authorized
		Key generation Management	
		Key export Management	

8.1.1.1 SCD/SVD Management

The TOE controls the access on every object it possesses, in particular the SCD and the SVD.

In phase 6, S.Admin is the personalization Agent, and as such always has the attribute "SCD/SVD Management" set to "Authorized".

In phase 7, two access modes may be distinguished by the TOE
- SCD/SVD generation (SSCD type 2)
- SCD/SVD import (SSCD type 3)

The access condition is granted to a user if the following conditions are met:
- The User is successfully authenticated
- The User was given the right to manage the SCD & SVD (import and/or generation).

If these two conditions are fulfilled, the attribute "SCD/SVD management" is set to "authorized", otherwise it is set to "not authorized".

8.1.1.2 SCD Operational

The attribute "SCD operational" is granted by the submission of the RAD by the User Signatory. The RAD may be a PIN or a Biometric template.

8.1.1.3 IAS ECC Management

The TOE may be configured to allow:
- communication in contact and/or contactless mode
- qualified signature to be computed from a hash off card
- enable/disable the authentication mechanism based on symmetric scheme
- enable/disable the authentication mechanism based on asymmetric scheme

The value of the related security attributes may be changed in phase 6 by the "Personalisation Agent" and in phase 7 by "TOE_Administrator".

8.1.1.4 Key Management

In phase 6, the Personalisation Agent has the attribute Key import, generation and export Management set to Authorized

In phase 7, the TOE controls the access on every object it holds, in particular Keys including Diffie Hellman Domain parameters.

The access condition is granted to a user if the following conditions are fulfilled:
- The Subject is successfully authenticated
- The Subject was given the right to import/generate/export a key

When these two conditions are fulfilled, the security attribute is set to authorized, otherwise it is set to not authorized

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[2]
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[3]
that meet the following:
(1) [ANSIX9.31]
(2) [IEEE][4]

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[5] that meets the following: none[6].

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 on demand of the Signatory or Administrator. The destruction of the SCD is mandatory before the SCD is re-imported 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".

[2] [assignment: cryptographic key generation algorithm]
[3] [assignment: cryptographic key sizes]
[4] [assignment: list of standards]
[5] [assignment: cryptographic key destruction method]
[6] [assignment: list of standards].

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.

Refinement:
In phase 6, S.User is the "Personalisation Agent" and always has the security attribute "SCD/SVD Management" set to "authorized".
In phase 7, depending on the use case, the role allowed to generate SCD/SVD may be restricted to R.Admin, one of its sub roles, to R.Sigy or any combination of them.

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.

Refinement:
In phase 6, S.User is the «Personalisation Agent» and always has the security attribute "SCD/SVD Management" set to "authorized".

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 or R.Sigy[7] is allowed to export SVD.

Refinement:
In phase 6 R.Admin is the "Personalisation Agent" and always has the security attribute "SCD/SVD Management" set to "authorized".
In phase 7, depending on the use case, the role allowed to export the SVD may be restricted to R.Admin, one of its sub roles, to "R.Sigy" or any combination of them.

[7] [selection : R.Admin, R.Sigy]

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.

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.

Refinement:
In phase 6, S.User is the "Personalisation Agent" and always has the security attribute "SCD/SVD Management" set to "authorized".
In phase 7, depending on the use case, the role allowed to import the SCD may be restricted to R.Admin, one of its sub roles, to R.Sigy or any combination of them.

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.

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, 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. SVD
3. RAD
4. Keys including Diffie hellman parameters

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[8].

Application note:
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.
In phase 6, the authorized CSP is the «Personalisation 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 **FIA_UID.1** *Timing of identification*

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

[8] [assignment: additional importation control rules]

FIA_UID.1.1	The TSF shall allow (1) Self-test according to FPT_TST.1 (2) establishing a trusted channel between the CSP and the TOE by means of the TSF required by FTP_ITC.1/SCD[9]. 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.

8.1.2.1.14 **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 CSP and the TOE by means of the TSF required by FTP_ITC.1/SCD[10]. 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.

8.1.2.1.15 **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 and TOE_Administrator.
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

8.1.2.1.16 **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, modification, and unblocking 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) Management of the TOE (8) Key management[11]

Application Note: There is no default value for the SCD Identifier

[9] [assignment : list of additional TSF-mediated actions]

[10] [assignment : list of additional TSF-mediated actions]

[11] [assignment : list of other security management functions to be provided by the TSF]

8.1.2.1.17 **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.

8.1.2.1.18 **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.
(3) IAS ECC Management
(4) Key Management

8.1.2.1.19 **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, IAS ECC Administration SFP, and Key Management 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.

Refinement:

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.Admin**
IAS ECC Management	**6&7**	**Personalisation Agent in phase 6 and TOE_Administrator in phase 7**
Key Management	**6&7**	**Personalisation Agent in phase 6 R.Sigy, CSP, SCA, HID, IFD and User_Admin in phase 7**

8.1.2.1.20 **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.Sigy successfully generates an SCD/SVD pair the security attribute "SCD operational of the SCD" shall be set to "yes" as a single operation. (3) 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 (4) 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

8.1.2.1.21 **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.

8.1.2.1.22 **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 [12]in excess of limits specified by the state of the art attacks on smart card IC[13] enabling access to RAD, SCD and Keys.
FPT.EMS.1.2	The TSF shall ensure all users[14] are unable to use the following interface external contacts emanations[15] to gain access to RAD, SCD, and Keys.

8.1.2.1.23 **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) card reset or tearing (3) Security violation detected by [PLT] with FAU_ARP.1, (4) Failure detected by [PLT] with FPT_FLS.1, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, and FPT_FLS.1/SCP (5) Integrity error detected on RAD, SCD, and Keys[16]

[12] [assignment : types of emissions]

[13] [assignment: specified limits]

[14] [assignment: type of users]

[15] [assignment: type of connection]

[16] [assignment : list of other types of failures in the TSF]

8.1.2.1.24 **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.

8.1.2.1.25 **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[17] to the TSF[18] by responding automatically such that the SFRs are always enforced.

8.1.2.1.26 **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, periodically during normal operation[19] 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.

8.1.2.1.27 **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[20]

[17] [assignment: physical tampering scenarios]

[18] [assignment: list of TSF devices/elements]

[19] [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-test should occur]]

[20] [assignment : list of other functions for which a trusted channel is required]

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 digital signature creation in accordance with a specified cryptographic algorithm:

- PKCS#1 V1.5 Block type 1 with Message Digest Info RSA CRT and hashing algorithm SHA-1 or SHA-256
- ECDSA-SHA1, SHA-224, SHA-256, SHA-384, SHA-512[21]

and cryptographic key sizes:

- RSA: 1024 bits or 1536 bits or 2048 bits
- ECDSA: Any elliptic curve from 160 bits up to 521 bits with prime field p[22]

that meet the following:

- [PKCS#1]
- [ANSIX9.62][23]

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.

---

[21] [assignment : cryptographic algorithm]
[22] [assignment : cryptographic key sizes]
[23] [assignment : list of standards]

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[24] 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 if the latter uses a RAD 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

[24] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]

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 the RAD to R.Sigy.

Refinement: This requirement applies only if the RAD belonging to S.Sigy.

8.1.3 Additional SFRs

8.1.3.1 Phase 6

8.1.3.1.1 **FCS_COP.1/GP secret data protection** *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[25] in accordance with a specified cryptographic algorithm: • SCP02 using TDES • SCP03 using AES • Proprietary SCP03 using AES[26] and cryptographic key sizes: • 128 bits • 128, 192, and 256 bits • 128,192, and 256 bits[27] that meet the following: • [GP2.2.1] • [SCP03] • [PLT][28]

Application Note 1:
The type of algorithm used by the TOE depends on the configuration set during the javacard open platform personalisation (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.

8.1.3.1.2 **FMT_MTD.1/TOE Serial Number** *Management of TSF data*

[25] [assignment : list of cryptographic operations]
[26] [assignment : cryptographic algorithm]
[27] [assignment : cryptographic key sizes]
[28] [assignment : list of standards]

Hierarchical to:	No other components.
Dependencies:	No dependencies

FMT_MTD.1.1/TOE Serial Number — The TSF shall restrict the ability to set[29] the serial number of the TOE[30] to Personalisation_Agent[31]

8.1.3.1.3 **FMT_MTD.1/TOE state** *Management of TSF data*

Hierarchical to:	No other components.
Dependencies:	No dependencies

FMT_MTD.1.1/TOE state — The TSF shall restrict the ability to switch[32] the TOE from phase 6 to phase 7[33] to Personalisation_Agent[34]

8.1.3.2 Phase 7

8.1.3.2.1 **FCS_CKM.1/Session keys** 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[35] and specified cryptographic key sizes:

(1) DES keys of 128 bits
(2) Two AES keys of 128, 192, and 256 bits
(3) Three AES keys of 128, 192, and 256 bits[36]

that meet the following: [14890][37]

8.1.3.2.2 **FCS_CKM.4/Session keys** *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/Session keys — The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting the buffer containing the key with zero[38] that meets the following: none[39].

[29] [selection : change_default, query, modify, delete, clear, [assignment : other operations]]
[30] [assignment : list of TSF data]
[31] [assignment : the authorized identified roles]
[32] [selection : change_default, query, modify, delete, clear, [assignment : other operations]]
[33] [assignment : list of TSF data]
[34] [assignment : the authorized identified roles]
[35] [assignment: cryptographic key generation algorithm]
[36] [assignment: cryptographic key sizes]
[37] [assignment: list of standards]
[38] [assignment: cryptographic key destruction method]
[39] [assignment: list of standards].

8.1.3.2.3 **FCS_COP.1/DH Computation** *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/DH Computation	The TSF shall perform Key Agreement[40] in accordance with a specified cryptographic algorithm: Diffie Hellmann[41] and cryptographic key sizes: 1024 bits, or 1536 bits, or 2048 bits[42] that meet the following: [PKCS#3][43]

8.1.3.2.4 **FCS_COP.1/SM in confidentiality** *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/SM in confidentiality	The TSF shall perform Secure Messaging in confidentiality[44] in accordance with a specified cryptographic algorithm: (1) Encryption with TDES EDE in CBC mode (2) Encryption with AES in CBC mode[45] and cryptographic key sizes: (1) 128 bits (2) 128 bits, 192 bits and 256 bits[46] that meet the following: [11568-2][47]

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

8.1.3.2.5 **FCS_COP.1/SM in integrity** *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/SM in integrity	The TSF shall perform Secure Messaging in integrity and authenticity[48] in accordance with a specified cryptographic algorithm: (1) Retail MAC: MAC algorithm 3 with padding method 2 and DES bloc Cipher

[40] [assignment : list of cryptographic operations]
[41] [assignment : cryptographic algorithm]
[42] [assignment : cryptographic key sizes]
[43] [assignment : list of standards]
[44] [assignment : list of cryptographic operations]
[45] [assignment : cryptographic algorithm]
[46] [assignment : cryptographic key sizes]
[47] [assignment : list of standards]
[48] [assignment : list of cryptographic operations]

(2) EMAC: MAC algorithm 2 with padding method 2 and AES bloc Cipher with a length of eight bytes
(3) CMAC: CMAC with pre padding method 2 and AES bloc Cipher with a length of eight bytes [49]

and cryptographic key sizes:
(1) 128 bits
(2) 128 bits, 192 bits and 256 bits
(3) 128 bits, 192 bits and 256 bits[50]

that meet the following:
(1) [9797-1]
(2) [9797-1]
(3) [SP800-38B][51]

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

8.1.3.2.6 **FCS_COP.1/C/S Auth** *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/C/S Auth	The TSF shall perform Client/Server Authentication[52] in accordance with a specified cryptographic algorithm: raw ECDSA[53] and cryptographic key sizes: Any elliptic curve from 160 bits up to 521 bits with prime field p[54] that meet the following: [ANSIX9.62][55]

8.1.3.2.7 **FCS_COP.1/Enc key decipherment** *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/Enc key decipherment	The TSF shall perform Encryption key decipherment[56] in accordance with a specified cryptographic algorithm: Diffie Hellman on an Elliptic curve[57] and cryptographic key sizes: Any elliptic curve from 160 bits up to 521 bits with prime field p[58] that meet the following: [TR03111][59]

8.1.3.2.8 **FCS_COP.1/Sym role Auth** *Cryptographic operation*

[49] [assignment : cryptographic algorithm]
[50] [assignment : cryptographic key sizes]
[51] [assignment : list of standards]
[52] [assignment : list of cryptographic operations]
[53] [assignment : cryptographic algorithm]
[54] [assignment : cryptographic key sizes]
[55] [assignment : list of standards]
[56] [assignment : list of cryptographic operations]
[57] [assignment : cryptographic algorithm]
[58] [assignment : cryptographic key sizes]
[59] [assignment : list of standards]

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/Sym role Auth

The TSF shall perform Symmetric role Authentication[60] in accordance with a specified cryptographic algorithm:

(1) Encryption using Triple DES EDE in mode CBC, Signature using Retail MAC
(2) Encryption using AES in mode CBC, Signature using EMAC
(3) Encryption using AES in mode CBC, Signature using CMAC
(4) Encryption using Triple DES EDE in CBC mode [61]

and cryptographic key sizes:

(5) 128 bits
(6) 128, 192, and 256 bits
(7) 128, 192, and 256 bits
(8) 128 bits[62]

that meet the following:

(1) [IASECC]
(2) [14890]
(3) [14890]
(4) [Minidriver][63]

8.1.3.2.9 **FCS_COP.1/Sym Device Auth** *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/Sym Device Auth

The TSF shall perform Symmetric Device Authentication[64] in accordance with a specified cryptographic algorithm:

(1) Encryption using Triple DES EDE in mode CBC, Signature using Retail MAC
(2) Encryption using AES in mode CBC, Signature using EMAC
(3) Encryption using AES in mode CBC, Signature using CMAC[65]

and cryptographic key sizes:

(1) 128 bits
(2) 128, 192, and 256 bits
(3) 128, 192, and 256 bits[66]

that meet the following:

(1) [IASECC]
(2) [14890]
(3) [14890][67]

[60] [assignment : list of cryptographic operations]
[61] [assignment : cryptographic algorithm]
[62] [assignment : cryptographic key sizes]
[63] [assignment : list of standards]
[64] [assignment : list of cryptographic operations]
[65] [assignment : cryptographic algorithm]
[66] [assignment : cryptographic key sizes]
[67] [assignment : list of standards]

8.1.3.2.10 **FCS_COP.1/Certificate verification** *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/Certificate verification The TSF shall perform Certificate verification[68] in accordance with a specified cryptographic algorithm: RSA with ISO/IEC 9796-2 padding with partial recovery and with SHA-1 or SHA-256[69] and cryptographic key sizes: 1024, 1536, or 2048 bits[70] that meet the following: [IASECC][71]

8.1.3.2.11 **FCS_COP.1/Asym Role Auth** *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/Asym Role Auth The TSF shall perform Asymmetric Role Authentication[72] in accordance with a specified cryptographic algorithm: RSA with ISO/IEC 9796-2 padding with partial recovery and with SHA-1 or SHA-256[73] and cryptographic key sizes: 1024, 1536, or 2048 bits[74] that meet the following: [IASECC][75]

8.1.3.2.12 **FCS_COP.1/Asym Internal DAPP Auth** *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/Asym Internal DAPP Auth The TSF shall perform Asymmetric Internal DAPP Authentication[76] in accordance with a specified cryptographic algorithm: RSA with ISO/IEC 9796-2 padding with partial recovery and with SHA-1 or SHA-256[77] and cryptographic key sizes: 1024, 1536, or 2048 bits[78] that meet the following: [IASECC][79]

8.1.3.2.13 **FCS_COP.1/Asym External DAPP Auth** *Cryptographic operation*

---

68 [assignment : list of cryptographic operations]
69 [assignment : cryptographic algorithm]
70 [assignment : cryptographic key sizes]
71 [assignment : list of standards]
72 [assignment : list of cryptographic operations]
73 [assignment : cryptographic algorithm]
74 [assignment : cryptographic key sizes]
75 [assignment : list of standards]
76 [assignment : list of cryptographic operations]
77 [assignment : cryptographic algorithm]
78 [assignment : cryptographic key sizes]
79 [assignment : list of standards]

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/Asym External DAPP Auth — The TSF shall perform Asymmetric External DAPP Authentication[80] in accordance with a specified cryptographic algorithm: RSA with ISO/IEC 9796-2 padding with partial recovery and with SHA-1 or SHA-256[81] and cryptographic key sizes: 1024, 1536, or 2048 bits[82] that meet the following: [IASECC][83]

8.1.3.2.14 **FMT_MTD.1/SCD and SCD ID** *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/SCD and SCD_ID

The TSF shall restrict the ability to select[84] the SCD using a SCD Identifier[85] to S.User[86].

Application note:
At creation, the SCD is given a SCD identifier that will be permanently associated to it and used by the TOE to select it.

8.1.3.2.15 **FMT_MTD.1/Unblock** *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/Unblock — The TSF shall restrict the ability to unblock[87] the RAD[88] to R.Admin[89].

Application note:
This SFR apply to any RAD (belonging to R.Sigy or R.Admin).

8.1.3.3 Phase 6 & 7

8.1.3.3.1 **FCS_CKM.1/Keys** Cryptographic key generation

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or

[80] [assignment : list of cryptographic operations]
[81] [assignment : cryptographic algorithm]
[82] [assignment : cryptographic key sizes]
[83] [assignment : list of standards]
[84] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[85] [assignment: list of TSF data]
[86] [assignment: The authorized identified roles]
[87] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[88] [assignment: list of TSF data]
[89] [assignment: The authorized identified roles]

FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/Keys — The TSF shall generate **Keys** in accordance with a specified cryptographic key generation algorithm:
(3) RSA key generation
(4) Key pair over Elliptic curve[90]
and specified cryptographic key sizes:
(3) 1024 bits or 1536 bits or 2048 bits
(4) Any elliptic curve from 160 bits up to 521 bits with prime field p[91]
that meet the following:
(3) [ANSIX9.31]
(4) [IEEE][92]

8.1.3.3.2 **FCS_COP.1/data hashing** *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/data hashing — The TSF shall perform data hashing[93] in accordance with a specified cryptographic algorithm: SHA-1, partial SHA-1, SHA-224, SHA-256, partial SHA-256, SHA-384 and SHA-512[94] and cryptographic key sizes: none[95] that meet the following: [FIPS 180-3][96]

8.1.3.3.3 **FCS_COP.1/GP Auth** *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 Auth — The TSF shall perform Mutual Authentication[97] in accordance with a specified cryptographic algorithm:
(1) SCP02 using TDES
(2) SCP03 using AES
(3) Proprietary SCP03 using AES[98]
and cryptographic key sizes:
(1) 128 bits
(2) 128, 192, and 256 bits
(3) 128,192, and 256 bits[99]
that meet the following:

[90] [assignment: cryptographic key generation algorithm]
[91] [assignment: cryptographic key sizes]
[92] [assignment: list of standards]
[93] [assignment : list of cryptographic operations]
[94] [assignment : cryptographic algorithm]
[95] [assignment : cryptographic key sizes]
[96] [assignment : list of standards]
[97] [assignment : list of cryptographic operations]
[98] [assignment : cryptographic algorithm]
[99] [assignment : cryptographic key sizes]

(1) [GP2.2.1]
(2) [SCP03]
(3) [PLT][100]

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 process the authentication. Their import/generation and destruction are managed by the platform.

8.1.3.3.4 **FCS_RNG.1** *Random Number Generation*

Hierarchical to: No other components.
Dependencies: No dependencies

FCS_RNG.1.1 The TSF shall provide a hybrid[101] random number generator that implements none[102].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [RGS_B1][103].

8.1.3.3.5 **FDP_ACC.1/IASECC Administration** *Subset access control*

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

FDP_ACC.1.1/IAS ECC Administration The TSF shall enforce the IAS ECC Administration SFP[104] on
(1) Subjects: TOE_Administrator (in phase 7), Personalisation Agent (Phase 6)
(2) objects: internal objects described in IASECC management
(3) operations: IAS ECC Management[105].

8.1.3.3.6 **FDP_ACC.1/key management** *Subset access control*

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

FDP_ACC.1.1/key management The TSF shall enforce the key management SFP[106] on
(4) Subjects: S.User
(5) objects:keys including Diffie Hellman Domain parameters
(6) operations:
- Import of keys and Diffie Hellman Domain parameters
- Generation of asymmetric key pair

[100] [assignment : list of standards]
[101] [selection : physical, non physical true, deterministic hybrid]
[102] [assignment : list of security capabilities]
[103] [assignment : a defined quality metric]
[104] [assignment : access control SFP]
[105] [assignment : list of subjects, objects and operations among subjects and objects covered by the SFP]
[106] [assignment : access control SFP]

- Export of public keys and Diffie Hellman Domain parameters[107].

8.1.3.3.7 **FDP_ACF.1/ IASECC Administration** *Security attribute based access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control

FDP_ACF.1.1/ IASECC Administration: The TSF shall enforce the IASECC Administration SFP[108] to objects based on the following: S.Admin is associated with the security attribute "IAS ECC Management" [109].

FDP_ACF.1.2/ IASECC Administration: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) In phase 6, subject with the security attribute "role" set to "Personalization Agent" is allowed to modify the IAS ECC Management attributes
(2) In phase 7, subject with the security attribute "role" set to "TOE_Administrator" is allowed to modify the IAS ECC Management attributes[110].

FDP_ACF.1.3/ IASECC Administration: The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[111].

FDP_ACF.1.4/ IASECC Administration: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) In phase 6, subject without the security attribute "role" set to "Personalization Agent" is allowed to modify the IAS ECC Management attributes
(2) In phase 7, subject without the security attribute "role" set to "TOE_Administrator" is allowed to modify the IAS ECC Management attributes[112].

8.1.3.3.8 **FDP_ACF.1/key management** *Security attribute based access control*

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control

FDP_ACF.1.1/ key management: The TSF shall enforce the key management SFP[113] to objects based on the following: S.User is associated with the security attribute "Key management"[114].

FDP_ACF.1.2/ key management: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) In phase 7, the user with the security attribute role set to S.Sigy, User_Admin, CSP, SCA, HID, IFD and with the security attribute

[107] [assignment : list of subjects, objects and operations among subjects and objects covered by the SFP]
[108] [assignment : access control SFP]
[109] [assignment : list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[110] [assignment : rules governing access among controlled subjects and controlled objects using controlled operationson controlled objects]
[111] [assignment : rules, based on security attributes, that explicitly authorise access of subjects to objects]
[112] [assignment : rules, based on security attributes, that explicitly deny access of subjects to objects]
[113] [assignment : access control SFP]
[114] [assignment : list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

"Key import Management" set to "authorised" is allowed to import key and Diffie Hellman Domain parameters
(2) In phase 6, the user with the security attribute "role" set to “Personalisation Agent” is allowed to import keys and Diffie Hellman Domain parameters
(3) In phase 7, the user with the security attribute role set to S.Sigy, User_Admin, CSP, SCA, HID, IFD and with the security attribute "Key generation Management" set to "authorised" is allowed to generate a key pair
(4) In phase 6, the user with the security attribute role set to Personalisation Agent is allowed to generate a key pair
(5) In phase 7, the user with the security attribute role set to S.Sigy, User_Admin, CSP, SCA, HID, IFD and with the security attribute "Key export Management" set to "authorised" is allowed to export a public key and Diffie Hellman Domain parameters
(6) In phase 6, the user with the security attribute "role" set to “Personalisation Agent” is allowed to export a public key and Diffie Hellman Domain parameters
(7) In phase 7, if the import, export or generation operation is set to Never, any user will not be allowed to perform the operation
(8) In phase 7, if the export operation is set to Always, any user will be allowed to perform the operation[115].

Application note:
In phase 6, the entity with the role “Personalisation Agent” always has the security attribute “Key export Management, “Key import Management”, and “Key generation Management” set to “authorized”.
In phase 7, depending on the use case, the “role” allowed to import, generate or export the keys may be restricted to R.Sigy, User_Admin, CSP, SCA, HID, IFD, or any combination of them.

FDP_ACF.1.3/ key management	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[116].
FDP_ACF.1.4/ key management	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[117].

8.1.3.3.9 **FDP_ETC.1/keys** *Export to Outside TSF control*

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

FDP_ETC.1.1/keys	The TSF shall enforce the key management SFP[118] when exporting user data, controlled under the SFP(s), outside of the TOE.
FDP_ETC.1.2/keys	the TSF shall export the user data without the user data’s associated security attributes.

8.1.3.3.10 **FDP_ITC.1/Keys** *Import of user data without security attributes*

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

[115] [assignment : rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[116] [assignment : rules, based on security attributes, that explicitly authorise access of subjects to objects]
[117] [assignment : rules, based on security attributes, that explicitly deny access of subjects to objects]
[118] [assignment : access control SFP]

FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialization

FDP_ITC.1.1/Keys — The TSF shall enforce the key management SFP[119] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2/Keys — The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3/Keys — The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: Keys shall be sent by the User with the "role" set to S.Sigy, User_Admin, Personalisation Agent, CSP, SCA, HID, IFD[120].

Application note:
In phase 7, depending on the use case, the "role" allowed to import, generate or export the keys may be restricted to R.Sigy, User_Admin, CSP, SCA, HID, IFD or any combination of them.

8.1.3.3.11 **FIA_AFL.1/Auth keys** *Authentication failure handling*

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

FIA_AFL.1.1/Auth keys — The TSF shall detect when **[selection: [assignment :positive integer number], an administrator configurable positive integer within 1 and 15]** unsuccessful authentication attempts occur related to consecutive failed authentication attempts.

FIA_AFL .1.2/Auth keys — When the defined number of unsuccessful authentication attempts has been met, the TSF shall **[assignment: List of actions]**

Refinements:

Type of entity	*Entity*	*Selection for FIA_AFL.1.1*	*list of actions*
User	"Personalisation Agent"	***Positive integer number '1'***	***Time of next authentication increases***
User	"TOE_Administrator"	***Positive integer number '1'***	***Time of next authentication increases***
User	"User_Admin" (when using symmetric role authentication)	***Administrator configurable positive integer 'N' 0 ≤N ≤15***	***If N= '0', no actions are taken. If N != '0', the key is blocked***
User	"User_Admin" (when using asymmetric role authentication)	***Positive integer number '1'***	***The key is deallocated with respect to FDP_RIP.1.1***
User	"CSP, SCA, HID, IFD" (when using symmetric device authentication)	***Administrator configurable positive integer 'N' 0 ≤N ≤15***	***If N= '0', no actions are taken. If N != '0', the key is blocked***
User	"CSP, SCA, HID, IFD" (when using asymmetric device authentication)	***Positive integer number '1'***	***The key is deallocated with respect to FDP_RIP.1.1***

[119] [assignment : access control SFP]
[120] [assignement :]

8.1.3.3.12 **FMT_MSA.1/ key management** *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/ key management

The TSF shall enforce the key management SFP[121] to restrict the ability to modify[122] the security attributes Key management[123] to:

(1) S.Sigy
(2) User_Admin
(3) Personalisation Agent
(4) CSP
(5) SCA
(6) HID
(7) IFD[124]

8.1.3.3.13 **FMT_MSA.1/ TOE management** *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/ TOE management

The TSF shall enforce the IASECC Administration SFP[125] to restrict the ability to modify[126] the security attributes IASECC Management[127] to:

(1) TOE_Administrator, or
(2) Personalisation Agent[128]

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

[121] [assignment : access control SFP]
[122] [selection: change_default, query, modidy, delete, [assignment: other operations]]
[123] [assignment : list of security attributes]
[124] [assignment : the authorized identified roles]
[125] [assignment : access control SFP]
[126] [selection: change_default, query, modidy, delete, [assignment: other operations]]
[127] [assignment : list of security attributes]
[128] [assignment : the authorized identified roles]

	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 1- 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.Authentication_Secure	OT.SCD/SVD_Management	OT.Key_Lifecycle_Security	OT.Keys_Secrecy	OT.TOE_AuthKey_Unique	OT.Lifecycle_Management	OT.eServices
FCS_CKM.1/SCD/SVD_Generation	X		X	X		X													
FCS_CKM.1/Keys															X	X	X		
FCS_CKM.1/Session_keys													X						
FCS_CKM.4	X					X									X	X			
FCS_CKM.4/Session keys													X						
FCS_COP.1/Sign	X						X												
FCS_COP.1/GP secret data protection								X					X						
FCS_COP.1/DH computation													X		X				X
FCS_COP.1/SM in confidentiality													X						
FCS_COP.1/SM in integrity													X						
FCS_COP.1/C/S Auth															X				X
FCS_COP.1/Enc key decipherment															X				X
FCS_COP.1/Sym role auth													X		X				X
FCS_COP.1/Sym Device auth													X		X				X
FCS_COP.1/Certificate verification													X		X				X
FCS_COP.1/Asym role auth													X		X				X
FCS_COP.1/Asym internal DAPP auth													X		X				X
FCS_COP.1/Asym external DAPPauth													X		X				X
FCS_COP.1/data hashing													X						

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.Authentication_Secure	OT.SCD/SVD_Management	OT.Key_Lifecycle_Security	OT.Keys_Secrecy	OT.TOE_AuthKey_Unique	OT.Lifecycle_Management	OT.eServices
FCS_COP.1/GP Auth								X					X					X	
FCS_RNG.1								X					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	
FDP_ACC.1/Signature_creation	X							X										X	
FDP_ACC.1/IASECC Administration													X					X	
FDP_ACC.1/key management													X		X			X	
FDP_ACF.1/SCD/SVD_Generation	X	X																X	
FDP_ACF.1/SVD_Transfer	X																	X	
FDP_ACF.1/SCD_import	X				X													X	
FDP_ACF.1/Signature_creation	X							X										X	
FDP_ACF.1/IASECC Administration													X					X	
FDP_ACF.1/key management													X		X			X	
FDP_RIP.1						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_ETC.1/keys															X	X			
FDP_ITC.1/keys															X	X			
FIA_AFL.1/RAD								X					X						X
FIA_AFL.1/Auth keys								X					X					X	X
FIA_UAU.1		X			X			X					X					X	X

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.Authentication_Secure	OT.SCD/SVD_Management	OT.Key_Lifecycle_Security	OT.Keys_Secrecy	OT.TOE_AuthKey_Unique	OT.Lifecycle_Management	OT.eServices
FIA_UID.1		X			X			X					X					X	X
FMT_MOF.1	X							X										X	
FMT_MSA.1/Admin	X	X																X	
FMT_MSA.1/Signatory	X							X											
FMT_MSA.1/key management															X			X	
FMT_MSA.1/TOE management													X					X	
FMT_MSA.2	X	X						X					X		X			X	
FMT_MSA.3	X	X						X					X		X			X	
FMT_MSA.4	X	X		X				X					X						
FMT_MTD.1/Admin	X							X										X	
FMT_MTD.1/Signatory	X							X										X	
FMT_MTD.1/TOE serial number																		X	
FMT_MTD.1/TOE state																		X	
FMT_MTD.1/SCD and SCD ID														X					
FMT_MTD.1/Unblock								X										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
FPT_FLS.1						X										X			
FPT_PHP.1											X								
FPT_PHP.3						X						X				X			
FPT_TST.1	X					X	X						X		X	X			
FTP_ITC.1/SCD	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_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 *ensures the unblocking of the RAD is made under the sole control of the administrator.*
In phase 6, the RAD (PIN or Biometric Data) may be loaded on the TOE by the Personalisation Agent as defined in ***FMT_SMF.1****. The Personalisation Agent is authenticated with a mutual authentication performed with* ***FCS_RNG.1*** *and* ***FCS_COP.1/GP Auth,*** *and is authenticated with* ***FMT_SMR.1****. Effort to bypass the access control by a frontal exhaustive attack is blocked by* ***FIA_AFL.1/Auth keys****. During the mutual authentication, a session encryption key is agreed between the TOE and the Personalisation 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 Personalisation 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**.

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.Authentication_Secure (*Secure authentication mechanisms*) is provided by the cryptographic algorithms specified by (1) **FCS_COP.1/DH Computation**, **FCS_COP.1/Certificate verification**, **FCS_COP.1/Asym Internal DAPP Auth**, **FCS_COP.1/Asym External DAPP Auth** and **FCS_RNG.1** for the mutual authentication based on an asymmetric scheme (DAPP), (2) **FCS_RNG.1** and **FCS_COP.1/Sym Device auth** for the mutual authentication based on symmetric scheme, (3) **FCS_RNG.1** and **FCS_COP.1/GP Auth** for the authentication of the personalisation agent and of the "TOE_Administrator", (4) **FCS_RNG.1** and **FCS_COP.1.1/Sym Role Auth** for the authentication of an entity based on a symmetric scheme, (5) **FCS_COP.1/Certificate verification** , **FCS_COP.1/Asym Role Auth**, and **FCS_RNG.1** for the authentication of an entity based on an asymmetric scheme. All these requirements ensure the cryptographic robustness of the authentication mechanisms.

The use of a challenge freshly generated by the TOE with **FCS_RNG.1** in theses authentication protocols ensures a protection against replay attacks when authenticating external entities. **FIA_AFL.1/Auth keys** ensures a correct detection and protection of authentication failure or exhaustive attacks. 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 ((1) and (2)). This trusted channel ensures authenticity, integrity and confidentiality of communication. **FCS_CKM.1/Session keys** and **FCS_COP.1/Data hashing** 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_RNG.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/session keys** 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 type of authentication scheme used by the TOE to authenticate the administrator or perform a mutual authentication may be controlled by the “TOE_Administrator”. It may enforce the TOE to allow the use of symmetric scheme ((2) and (4)) and/or asymmetric ((1) and (5)) schemes. The TSF specified by **FIA_UID.1** and **FIA_UAU.1** provide “TOE_Administrator” identification and authentication prior to enabling access to authorised functions. The attributes of the authenticated “TOE_Administrator” are provided by **FMT_MSA.1/TOE Management**, **FMT_MSA.2**, **FMT_MSA.3 and FMT_MSA.4** for static attribute initialisation. Access control is provided by **FDP_ACC.1/IAS ECC Administration, FDP_ACF.1/ IAS ECC Administration**, **FMT_SMR.1** and **FMT_SMF.1**. Effort to bypass the access control by a frontal exhaustive attack is blocked by **FIA_AFL.1/Auth keys**.

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 as mandated by **FDP_ACF.1/ Key Management** and **FDP_ACC.1/ Key Management.** 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** and **FIA_AFL.1/Auth keys**.

OT.SCD/SVD_Management (*Management of SCD/SVD*) enforces the link between SCD and the matching certificate. This objective is ensured by **FMT_MTD.1/SCD and SCD_ID** that guarantees and unambiguous link between the SCD and its identifier, which is connected to the certificate.

OT.Key_LifeCycle_Security (*Lifecycle security of the key(s) stored in the TOE*)
The keys management is controlled by TSF according to **FDP_ACC.1/Key management**, **FDP_ACF.1/Key management**. Keys import is controlled by TSF according to **FDP_ITC.1/Keys** and keys export is controlled by TSF according to **FDP_ETC.1/Keys**. Secure Keys generation is ensured by **FCS_CKM.1/Keys**.

The secure keys usage is ensured cryptographically according to **FCS_COP.1/DH Computation**, **FCS_COP.1/C/S Auth**, **FCS_COP.1/Enc key Decipherement**, **FCS_COP.1/Sym Role Auth**, **FCS_COP.1/Asym Role Auth**, **FCS_COP.1/Sym Device Auth**, **FCS_COP.1/Certificate verification**, **FCS_COP.1/Asym internal DAPP Auth**, **FCS_COP.1/Asym external DAPP Auth**. Keys usage is controlled by access control **FDP_ACC.1/Keys management**, **FDP_ACF.1/Keys management** which is based on the security attribute secure TSF management according to **FMT_MSA.1/Key management**, **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.
FDP_ITC.1/Keys controls the key(s) import and **FDP_ETC/Keys** controls the key(s) export.
FCS_CKM.1/Keys 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 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/Keys**.

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 Personalisation Agent. The Personalisation Agent is authenticated with a mutual authentication performed with **FCS_RNG.1** and **FCS_COP.1/GP Auth** and is authenticated with **FMT_SMR.1.** Effort to bypass the access control by a frontal exhaustive attack is blocked by **FIA_AFL.1/Auth keys**.

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_ACC.1/IAS ECC Administration**, **FDP_ACC.1/Key Management**, **FDP_ACF.1/Signature creation**, **FDP_ACF.1/SVD transfer**, **FDP_ACF.1/SCD/SVD_Generation**, **FDP_ACF.1/SCD import**, **FDP_ACF.1/IAS ECC Administration, FDP_ACF.1/Key Management, FMT_MTD.1/Unblock, 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_ACC.1/IAS ECC Administration**, **FDP_ACC.1/Key Management**, **FDP_ACF.1/SVD transfer**, **FDP_ACF.1/SCD/SVD_Generation**, **FDP_ACF.1/SCD import**, **FDP_ACF.1/IAS ECC Administration, FDP_ACF.1/Key Management, FDP_ETC.1/Keys.** It may as well change TOE State (**FMT_MTD.1/TOE State**), load the serial number of the TOE (**FMT_MTD.1/TOE Serial Number).** These functions are protected by the Personalisation 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.1/TOE Management, FMT_MSA.1/Key Management**, **FMT_MSA.2**, **FMT_MSA.3** ensure that the sole Personalisation Agent can realize these functions. Effort to bypass the access control by a frontal exhaustive attack is blocked by **FIA_AFL.1/Auth keys**.

OT.eServices (*Provision of eServices*) is provided by the cryptographic mechanisms specified by (1) **FCS_COP.1/DH Computation**, (2) **FCS_COP.1/Certificate verification**, (3) **FCS_COP.1/C/S Auth**, (4) **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**. The Agent(s) entitled to perform such operations shall be authenticated with **FMT_SMR.1** using cryptographic protocols (1) **FCS_COP.1/DH Computation**, **FCS_COP.1/Certificate verification**, **FCS_COP.1/Asym Internal DAPP Auth**, **FCS_COP.1/Asym External DAPP Auth,** and **FCS_RNG.1** for the mutual authentication based on an asymmetric scheme (DAPP), (2) **FCS_RNG.1** and **FCS_COP.1/Sym Device auth** for the mutual authentication based on symmetric scheme, (3)

FCS_RNG.1 and **FCS_COP.1/Sym Role Auth** for the authentication of an entity based on a symmetric scheme, (4) **FCS_COP.1/Certificate verification** , **FCS_COP.1/Asym Role Auth**, and **FCS_RNG.1** for the authentication of an entity based on an asymmetric scheme. 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** and **FIA_AFL.1/Auth keys**.

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/Keys	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1/C/S Auth FCS_COP.1/Enc Key Decipherment FCS_COP.1/Certificate verification FCS_COP.1/Asym Internal DAPP Auth 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/Session Keys
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.1/SCD FDP_ITC.1/Keys FCS_CKM.1/SCD/SVD_Generation FCS_CKM.1/Keys
FCS_CKM.4/Session keys	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1/Session Keys
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/DH Computation	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys 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/data hashing	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_COP.1/C/S Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys FCS_CKM.4
FCS_COP.1/Enc key decipherment	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys FCS_CKM.4
FCS_COP.1/Sym role Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys FCS_CKM.4
FCS_COP.1/Sym Device Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys FCS_CKM.4

Functional Requirement	Dependencies	Satisfied by
FCS_COP.1/Certificate verification	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_COP.1/Asym Role Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_COP.1/Asym Internal DAPP Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FDP_ITC.1/Keys FCS_CKM.1/Keys FCS_CKM.4
FCS_COP.1/Asym External DAPP Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_COP.1/GP secret data protection	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_COP.1/GP Auth	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Not satisfied (See §8.3.3.2)
FCS_RNG.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/IASECC Administration	FDP_ACF.1	FDP_ACF.1/IASECC Administration
FDP_ACC.1/Key management	FDP_ACF.1	FDP_ACF.1/Key management
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/IASECC Administration	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/IASECC Administration FMT.MSA.3
FDP_ACF.1/Key management	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Key management 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_ITC/Keys	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	FDP_ACC.1/Key management 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
FDP_ETC/Keys	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1/Keys management
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	No dependencies	n/a
FIA_AFL.1/RAD	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/Auth keys	FIA_UAU.1	FIA_UAU.1
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_SMF.1	No dependencies	n/a
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1

Functional Requirement	Dependencies	Satisfied by
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.1/Key management	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	FDP_ACC.1/Key management, FMT_SMR.1, FMT_SMF.1
FMT_MSA.1/TOE management	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	FDP_ACC.1/IASECC Administration, 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_MSA.1/Key management FMT_MSA.1/TOE management 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/Unblock	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/SCD and SCD ID	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/TOE Serial Number	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/TOE state	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FPT_EMS.1	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

Table 2 - 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_ARC.1 ADV_FSP.4 ADV_TDS.3

	ADV_IMP.1 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1	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 3 - Satisfaction of dependencies of SAR

8.3.3.2 Justifications for non satisfaction of dependencies

FCS_COP.1/data hashing: The cryptographic algorithms SHA-1 and SHA-256 do not use any cryptographic key. Therefore none of the SFRs listed in the dependencies ([FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4) are needed to be defined for this specific instantiation of FCS_COP.1.

FCS_COP.1/Certificate verification : Two situations occur.

1- During the first round of certificate verification, the TOE uses a Root certificate verification public key. When using this key, the following dependencies apply FDP_ITC.1/Keys and FCS_CKM.4.

As this certificate verification public key may be generated by the TOE, the following dependency applies: FCS_CKM.1/Keys

The certificate contains an ephemeral public key protected by a cryptogram that only the certificate verification public key can check.

Upon successful verification of the certificate (ensured by FCS_COP.1.1 / Certificate Verification), the ephemeral public key nested within the certificate is securely imported in the TOE for the next use

2- In next step(s), the certificate is verified with the ephemeral key (which is extracted from a former certificate verification step).The certificate contains a public key protected by a cryptogram that only the certificate verification public key (which is trusted) can check.

Upon successful verification of the certificate (ensured by FCS_COP.1.1 / Certificate Verification), the key nested within the certificate (which is an ephemeral key) is securely imported in the TOE for the next use
When the certificate verification fails, or when the sequence for certificate verification fails, the ephemeral public key is erased with FDP_RIP.1.

FCS_COP.1/Asym Role Auth and FCS_COP.1/Asym External DAPP Auth: The key used for authentication is an ephemeral key. It is securely imported on the TOE through successful certificate verification (ensured by FCS_COP.1.1 / Certificate Verification) and by the initial link of trust coming from the Root Certificate verification public key, the following dependencies apply: FDP_ITC.1/Keys and FCS_CKM.4.

When the User Authentication fails, or when the sequence for authentication is not fulfilled, the ephemeral public key is erased with FDP_RIP.1.

FCS_COP.1/GP auth and FCS_COP.1/GP secret data protection:

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 via FDP_CKM.1 and FCS_CKM.4

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.RAD_MGT

This security function is involved in the management of the RAD, whether it is PIN or Biometric based. It ensures the link between each RAD(s) and its associated role (S.Sigy and S.Admin).

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

- In phase 6, it only allows the RAD(s) to be created by the Personalisation Agent. It requires the RAD to be encrypted in order to ensure its confidentiality. This security function ensures the Personalisation Agent can not verify the RAD, and impersonate the role R.Sigy.
- In phase 7, it only allows the RAD(s) to be created by R.Admin. Once loaded, the RAD can only be changed under control of R.Sigy and unblocked by the R.Admin.
- In phase 7, it allows the TOE to authenticate any Role using a RAD comparison (R.Sigy, and R.Admin if it uses a RAD).

This security function manages the validation process of the role associated to the RAD (R.Sigy or S.Admin). It performs the comparison of the VAD with the RAD, and upon successful comparison it authenticates the associated role. Each RAD is associated to an error counter which aims at ensuring its protecting against brute force attacks. Upon each submission of an incorrect VAD, it decrements the error counter, and restores it to its maximum value upon a successful VAD submission. When the error counter has reached ‘00’, the security function blocks the usage of the RAD, and in particular bans the authentication of the associated role, and the ability to change the RAD value (for both R.Sigy and R.Admin). Once blocked, the security function allows the unblocking of the RAD after the successful authentication of R.Admin (please note that the R.Admin required to unblock the RAD may be different from the one associated to the blocked RAD if ever).

This security function also ensures secure deallocation of VAD after verification and RAD after update.

This security function allows managing the RAD either through APDU commands, or through shared interfaces (using sharing mechanism). They enable other applets potentially present on the javacard platform to manage the RAD. The security function ensures the same security policy is applied on both interfaces, so that there are no logical backdoor on the RAD management.

This security function relies on SF.DEV_AUTH and SF.ADM_AUTH to authenticate R.Admin required to create the RAD.

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 enables to select the signature key to be used for the signature creation among all the signature key hold by the TOE.

The security function ensures the data hashing (if hash on card, or partial hashing is used), and the secure signature computation using either a RSA or ECDSA private key (SCD). During the signature creation, the coherency with the matching signature public key (SVD) is verified.

This security function relies on:
- SF.DEV_AUTH to establish a trusted channel with the SCA
- SF.RAD_MGT to authenticate the Signatory
- SF.SM to transmit the DTBS

9.1.3 SF.DEV_AUTH

This security function manages the device authentication between the TOE and an external entity.

The device authentication is a mutual authentication between the TOE and an external entity that may be either realized using symmetric or asymmetric cryptography. Upon successful mutual authentication, the security function computes a shared secret (called the seed) from random numbers generated by both the TOE and the external entity and known only to them. The seed is then used by SF.SM to generate session keys to protect communication in integrity, authenticity and confidentiality, and then maintain the trusted channel. As such, this security function allows generating a trusted channel with an external entity.

This security function allows the mutual authentication with the following external entities:
- Personalisation Agent (phase 6)
- SCA (phase 6 & 7), mingled with the personalisation agent in phase 6
- CSP (phase 6 & 7), mingled with the personalisation agent in phase 6
- HID (phase 6 & 7), mingled with the personalisation agent in phase 6
- IFD (phase 7)

It authenticates also the SSCD and proves its identity.

This security function manages as well the validation process of the role associated to the authentication key used by the trusted IT entity. Upon successful device authentication, the associated role is authenticated. Each key is associated to an error counter (it may be infinite) which aims at ensuring its protecting against brute force attacks. Upon each wrong authentication, it decrements the error counter (if present), and restores it to its maximum value upon a successful authentication. When the error counter has reached ‘00’, the security function blocks the usage of the key, and bans the authentication of the associated role.

9.1.4 SF.ADM_AUTH

This security function manages the authentication of external entities by the TOE. It is only active in phase 7.

This security function enables the TOE to authenticate external entities and may be either realized using symmetric or asymmetric cryptography.

This security function manages as well the validation process of the role associated to the authentication key used by the external entity. Upon successful authentication, the associated role is authenticated. Each key is associated to an error counter (it may be infinite) which aims at ensuring its protecting against brute force attacks. Upon each wrong authentication, it decrements the error counter (if present), and restores it to its maximum value upon a successful authentication. When the error counter has reached ‘00’, the security function blocks the usage of the key, and bans the authentication of the associated role.

This security function allows the authentication of the following roles:
- TOE_Administrator
- User_Admin

9.1.5 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.

This security function requires the TOE and the external entity to establish first a trusted channel using a device authentication (mutual) with SF.DEV_AUTH.

It ensures the following properties:

- In phase 6, it maintains the confidentiality, integrity and authenticity of the private keys (including the SCD), the symmetric keys (DES and AES), and the RAD (PIN and biometric template)
- In phase 6, it maintains the integrity and authenticity of the asymmetric public key (including the SVD) when being exported to the outside
- In phase 7, it maintains the confidentiality, integrity and authenticity of communication exchanged between the TOE and the external entity.

In phase 7, the confidentiality, integrity and authenticity of data is ensured by cryptographic means based on symmetric cryptography. Data are encrypted and signed using the symmetric session keys generated from the seed agreed during the device (mutual) authentication (see SF.DEV_AUTH). Moreover, the protection against replay attacks is ensured by the signature which is computed using a dynamic ICV, incremented at each new command.

In phase 6, the confidentiality (for the SCD), integrity and authenticity (for the SVD), is ensured by cryptographic means based on symmetric cryptography. Data are encrypted using the symmetric session keys generated from the seed agreed during the device (mutual) authentication (see SF.DEV_AUTH). The integrity of the SVD is ensured by the

This security function is also in charge of building the session keys from the seed computed by SF.DEV_AUTH. These session keys are ephemeral and unique, as the seed is computed from random numbers generated by the TOE and the external entity.

This security function is also in charge of destroying the session keys in case an error is detected (data not authentic or not integer), or when a command in plan text is sent.

9.1.6 SF.KEY_MGT

This security function is involved in the management of the 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 and SVD, and the DH parameters) to be loaded, generated and exported (for the public keys) by the Personalisation Agent. It also requires the private and secret keys to be encrypted in order to ensure their confidentiality. This security function ensures the Personalisation Agent can not use the keys it has loaded or generated. It ensures the personalisation Agent can not impersonate the associated role (in case of authentication keys), or create a signature with the SCD.
- In phase 7, it enforces access control over the management operations on the SCD and SVD (import, generation and export) and ensures the SCD is loaded in an encrypted form to ensure its confidentiality.
- In phase 7, it enforces access control over the management operations on the authentication and eServices keys (import, generation, and export of public keys) and the DH parameters (loading). It ensures that any loading, generation or public export operation is performed by an authenticated entity (Signatory, IFD, SCA, CSP, User_Admin), according to the TOE configuration.

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

This security function relies on:

- SF.DEV_AUTH to establish the trusted channel with the SSCD type 1
- SF.RAD_MGT to authenticate the Signatory
- SF.DEV_AUTH and SF.ADM_AUTH to authenticate the roles entitled to perform the operations
- SF.SM to maintain the trusted channel and transmit the DTBS

9.1.7 SF.CONF

This security function manages the configuration of the TOE.

1) It allows the modification of the following TOE attributes in both phase 6 and 7:

- Communication medium : contact and/or contactless
- Type of cryptography to be used for the external entities and subject authentication (symmetric or asymmetric)
- Type of DTBS to be used: the DTBS representation fully computed outside the TOE may be used

This security function ensures their initialization to a default values when the applet instance is created, and apply an access control over modification. Only the successfully authenticated Personalisation Agent (in phase 6) or "TOE_Administrator" (phase 7) can modify these attributes.

2) It also allows the modification of the following TOE attributes in phase 6:

- TOE serial number
- TOE State

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

3) It also allows the modification in phase 5 of the ability to retrieve the identification data of the TOE. The security function ensures an access control over this operation. Only the successfully authenticated Manufacturing Agent (phase 5) can modify these attributes.

This security function relies on

- SF.DEV_AUTH to authenticate the role personalisation Agent
- SF.ADM_AUTH to authenticate the role TOE_Administrator

9.1.8 SF.ESERVICE

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

This security function offers the following electronic services:

- C/S authentication
- Decryption key decipherment
- Certificate verification

9.1.9 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 (RAD, keys, DTBS) by performing selftests. 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.

9.1.10 SF.PHYS

This security function ensures the protection of the TOE against physical manipulation aiming at getting access to its assets. In particular, it ensures that the TOE

- detects physical manipulation (I/O manipulation, EM perturbation, temperature perturbation,…) and takes countermeasures.
- is protected against probing and that there is no information leakage that may be used to reconstruct sensitive data

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

10 Annex A – Composition with the underlying javacard platform

This annex discusses the composition with the underlying javacard platform [PLT] according to [JIL-COMP].

10.1 Evaluation assurance Level

The underlying javacard open platform [PLT] is certified at level EAL 5 augmented with AVA_VAN.5 and ALC_DVS.2. As such it complies with the Evaluation level assurance of the composite TOE.

10.2 Coverage of the Assumptions of the Javacard Open Platform (A.PLT vs TOE)

Assumption of [PLT]	Rationale
A.APPLET	No Native API can be nested within Applet loaded in post issuance as the platform only allows the loading of javacard bytecode
A.VERIFICATION	This assumption is upheld by OE.VERIFICATION and OE.CODE_EVIDENCE. These objectives are fulfilled by the applet included in the TOE (see §10.4). For the other applet that may be loaded on the TOE, it is covered by recommendations in [AGD_OPE] whose fulfilment shall be verified by the risk manager.

10.3 Coverage of the OSP of the Javacard Open Platform (OSP.PLT vs TOE)

OSP of [PLT]	Rationale
OSP.VERIFICATION	This OSP is upheld by OE.VERIFICATION and OE.CODE_EVIDENCE. These objectives are fulfilled by the applet included in the TOE (see §10.4). For the other applet that may be loaded on the TOE, it is covered by recommendations in [AGD_OPE] whose fulfilment shall be verified by the risk manager.

10.4 Coverage of the security objective of the Javacard Open Platform Environment (OE.PLT vs TOE)

Security objective of [PLT] environment	Rationale
OE.APPLET	No Native API can be nested within Applet loaded in post issuance as the platform only allows the loading of javacard bytecode
OE.VERIFICATION	This objective is fulfilled by the applet included in the TOE. The applet is verified then included in the generated platform in IDEMIA facilities. For the other applet that may be loaded on the TOE, it is covered by recommendations in [AGD_OPE] whose fulfilment shall be verified by the risk manager.
OE.CODE-EVIDENCE	The verified applet is included in the TOE pre-issuance in IDEMIA facilities then the whole TOE is sent to the manufacturer according to audited organizational measures. It ensures that it has not been changed since the code verification required in OE.VERIFICATION.

10.5 Support of the TOE TSFs by the Javacard Open Platform TSFs (TSF.TOE vs TSF.SFR)

The following table shows how the security functions of the Composite TOE are supported by the security functions of the underlying javacard open platform:

TSF of the TOE	Supported by the following TSF of [PLT]
SF.RAD_MGT	SF_CARDHOLDER_VERIFICATION
SF.SIG	SF_ENCRYPTION_AND_DECRYPTION SF_KEY_ACCESS SF_MESSAGE_DIGEST

	SF_RANDOM_NUMBER SF_SIGNATURE
SF.DEV_AUTH	SF_ENCRYPTION_AND_DECRYPTION SF_ENTITY_AUTHENTICATION/SECURE_CHANNEL SF_KEY_ACCESS SF_MESSAGE_DIGEST SF_RANDOM_NUMBER SF_SIGNATURE
SF.ADM_AUTH	SF_ENCRYPTION_AND_DECRYPTION SF_ENTITY_AUTHENTICATION/SECURE_CHANNEL SF_KEY_ACCESS SF_MESSAGE_DIGEST SF_RANDOM_NUMBER SF_SIGNATURE
SF.SM	SF_ENCRYPTION_AND_DECRYPTION SF_ENTITY_AUTHENTICATION/SECURE_CHANNEL SF_KEY_ACCESS SF_KEY_DISTRIBUTION SF_MESSAGE_DIGEST SF_SIGNATURE
SF.KEY_MGT	SF_KEY_ACCESS SF_KEY_DESTRUCTION SF_KEY_GENERATION
SF.CONF	SF_PREPERSONALISATION
SF.ESERVICE	SF_ENCRYPTION_AND_DECRYPTION SF_KEY_ACCESS SF_KEY_AGREEMENT SF_MESSAGE_DIGEST SF_SIGNATURE
SF.SAFESTATE_MGT	SF_ATOMIC_TRANSACTION SF_CLEARING_OF_SENSITIVE_INFORMATION SF_DATA_COHERENCY SF_DATA_INTEGRITY SF_EXCEPTION SF_FIREWALL SF_KEY_DESTRUCTION SF_KEY_MANAGEMENT SF_MEMORY_FAILURE SF_RUNTIME_VERIFIER SF_SIGNATURE SF_ENCRYPTION_AND_DECRYPTION
SF.PHYS	SF_HARDWARE_OPERATING SF_SIGNATURE SF_ENCRYPTION_AND_DECRYPTION SF_UNOBSERVABILITY

10.6 Support of the TOE SFRs by the Javacard Open Platform SFRs (SFR.TOE vs SFR.PLT)

The following table shows how the SFRs of the Composite TOE are supported by the SFRs of the underlying javacard open platform:

SFRs of the TOE	Supported by [PLT]	SFRs of [PLT]
FCS_CKM.1/SCD/SVD_Generation	Fully	FCS_CKM.1
FCS_CKM.1/Keys	Fully	FCS_CKM.1
FCS_CKM.1/Session Keys	N/A	N/A
FCS_CKM.4	Fully	FCS_CKM.4
FCS_CKM.4.1 / Session keys	Fully	FCS_CKM.4

		FDP_RIP.1/TRANSIENT ensures destruction of session keys upon card reset
FCS_COP.1/Sign	Partially	FCS_COP.1
FCS_COP.1/DH Computation	Fully	FCS_COP.1
FCS_COP.1/SM in confidentiality	Fully	FCS_COP.1
FCS_COP.1/SM in integrity	Fully	FCS_COP.1
FCS_COP.1/data hashing	Fully	FCS_COP.1
FCS_COP.1/C/S Auth	Partially	FCS_COP.1
FCS_COP.1/Enc key decipherment	Fully	FCS_COP.1
FCS_COP.1/Sym role Auth	Partially	FCS_COP.1
FCS_COP.1/Sym Device Auth	Partially	FCS_COP.1
FCS_COP.1/Certificate verification	Partially	FCS_COP.1
FCS_COP.1/Asym Role Auth	Partially	FCS_COP.1
FCS_COP.1/Asym Internal DAPP Auth	Partially	FCS_COP.1
FCS_COP.1/Asym External DAPP Auth	Partially	FCS_COP.1
FCS_COP.1/GP secret data protection	Fully	FCS_COP.1/CM
FCS_COP.1/GP Auth	Fully	FCS_COP.1/CM
FCS_RNG.1	Fully	FCS_RNG.1/SCP
FDP_ACC.1/SCD/SVD_Generation	N/A	N/A
FDP_ACC.1/SCD_Import	N/A	N/A
FDP_ACC.1/SVD_Transfer	N/A	N/A
FDP_ACC.1/Signature_Creation	N/A	N/A
FDP_ACC.1/IASECC Administration	N/A	N/A
FDP_ACC.1/Key management	N/A	N/A
FDP_ACF.1/SCD/SVD_Generation	N/A	N/A
FDP_ACF.1/SCD_Import	N/A	N/A
FDP_ACF.1/SVD_Transfer	N/A	N/A
FDP_ACF.1/Signature_Creation	N/A	N/A
FDP_ACF.1/IASECC Administration	N/A	N/A
FDP_ACF.1/Key management	N/A	N/A
FDP_RIP.1	Partially	FDP_RIP.1/OBJECTS FDP_RIP.1/ABORT FDP_RIP.1/APDU FDP_RIP.1/bArray FDP_RIP.1/KEYS FDP_RIP.1/TRANSIENT FDP_RIP.1/ADEL FDP_RIP.1/ODEL
FDP_SDI.2/Persistent	Fully	FDP_SDI.2 for the PINs, Biometric templates, keys and data stored in a secure store object FCS_CKM.3 provides access to the signature key used for integrity control of DH parameters FCS_COP.1 provides cryptographic means for the signature computation/verification of DH parameters
FDP_SDI.2/DTBS	Partially	FCS_CKM.3 provides access to the signature key used for integrity control FCS_COP.1 provides cryptographic means for the signature computation/verification
FDP_ITC.1/SCD	Partially	FCS_CKM.2 FCS_CKM.3
FDP_ITC/Keys	Partially	FCS_CKM.2 FCS_CKM.3
FDP_UCT.1/SCD	Partially	FCS_CKM.3

FDP_ETC/Keys	Partially	FCS_CKM.3
FIA_UAU.1	N/A	N/A
FIA_UID.1	N/A	N/A
FIA_AFL.1/RAD	Fully	FIA_AFL.1/PIN for the PIN FIA_AFL.1/PIN_BIO for the Biometric template
FIA_AFL.1/Auth keys	Partially	FIA_AFL.1/CM for the authentication of the roles Personalisation Agent and TOE_Administrator
FMT_SMR.1	N/A	N/A
FMT_SMF.1	Partially	FCS_CKM.3 for the use of the cryptographic keys
FMT_MOF.1	N/A	N/A
FMT_MSA.1/Admin	N/A	N/A
FMT_MSA.1/Signatory	N/A	N/A
FMT_MSA.1/Key management	N/A	N/A
FMT_MSA.1/TOE management	N/A	N/A
FMT_MSA.2	Partially	FDP_RIP.1/TRANSIENT ensures the security attributes stored in ephemeral memory (transient) are reset to restrictive default value after card reset.The following security attributes are concerned: -SCD/SVD management -SCD operational -Key management
FMT_MSA.3	Partially	FDP_RIP.1/TRANSIENT ensures the security attributes stored in ephemeral memory (transient) are reset to restrictive default value after card reset.The following security attributes are concerned: -SCD/SVD management -SCD operational -Key management
FMT_MSA.4	N/A	N/A
FMT_MTD.1/Admin	N/A	N/A
FMT_MTD.1/Signatory	Partially	FMT_MTD.1/PIN for the PIN change featureFMT_MTD.1/PIN_BIO for the Biometric template change feature
FMT_MTD.1/Unblock	Partially	FMT_MTD.1/PIN for the PIN unblocking feature FMT_MTD.1/PIN_BIO for the Biometric template unblocking feature
FMT_MTD.1/SCD and SCD ID	N/A	N/A
FMT_MTD.1/TOE Serial Number	N/A	N/A
FMT_MTD.1/TOE State	N/A	N/A
FPT_EMS.1	Fully	FPR_UNO.1 for the PIN and Biometric management FPR_UNO.1/Key_CM for the import/update of the authentication keys of the Personalisation Agent and TOE_Administrator FPR_UNO.1/USE_KEY when using the authentication keys of the Personalisation Agent and TOE_Administrator FPR_UNO.1/Applet when comparing two bytes arrays
FPT_FLS.1	Partially	FDP_ROL.1/FIREWALL ensures the TOE returns in a safe state in case an error occurs in an atomic transation FAU_ARP.1 ensures security actions are taken upon security violations and that the TOE returns in a safe state

		FPT_FLS.1 FPT_FLS.1/ADEL FPT_FLS.1/ODEL FPT_FLS.1/SCP FPT_RCV.3/SCP ensures automated recovery in the secure initial state FPT_RCV.4/SCP ensures a secure state in case of power loss during a reading/writing
FPT_PHP.1	Fully	FPT_PHP.3/SCP
FPT_PHP.3	Fully	FPT_PHP.3/SCP
FPT_TST.1	Partially	FPT_TST.1 FDP_SDI.2 for the integrity of patch and javacard packages. Any loss of integrity is detected
FTP_ITC.1/SCD	Partially	FTP_ITC.1/CM

10.7 Coverage of the composite ST threats by the platform threats

TOE threat	composite ST threat	Platform threat covering the Composite ST threat
T.SCD_Divulg	YES	This threat addresses the disclosure of the SCD during its generation or import which is covered by the applet but also by collecting information after its destruction, during storage or use which is directed against the platform and meet T.CONFID_APPLI_DATA, T.RESSOURCES, T.OBJ_DELETION, and T.PHYSICAL
T.SCD_Derive	YES	This threat addresses the derivation of the SCD from the SVD, the signature or any other publicly known data. A part of this threat involves the integrity of the SCD and is addressed against the platform. It meets T.INTEG_APPLI_DATA
T.Hack_Phys	YES	This threat is mainly addressed against the platform and a large part is covered by T.PHYSICAL. The following threats of the platform also cover it T.RESSOURCES, T.OBJ_DELETION, T.CONFID_APPLI_DATA and T.INTEG_APPLI_DATA
T.SVD_Forgery	YES	This threat addresses the forgery of the SVD. When the applet is responsible of its integrity while transporting it, the platform is responsible of its integrity inside the container. The platform threat for this problem is T.INTEG_APPLI_DATA
T.SigF_Misuse	NO	The platform is not involved in the protection of the TOE against this threat
T.DTBS_Forgery	NO	The applet ensures the integrity of the DTBS.
T.Sig_Forgery	YES	This threat addresses the forgery of the signature created by the TOE. A part of it is covered by the platform threat T.INTEG_APPLI_DATA (against SCD integrity)
T.Key_Divulg	YES	This threat addresses the disclosure of the authentication and eServices keys when transporting them which is covered by the applet but also by collecting information after their destruction, during storage or use which is directed against the platform and meet T.CONFID_APPLI_DATA, T.RESSOURCES, T.OBJ_DELETION, and T.PHYSICAL
T.Key_Derive	YES	This threat addresses the derivation of the authentication and eServices keys from the public keys, the authentication cryptogram or any other publicly known data. A part of this threat is under the responsibility of the platform and meets T.INTEG_APPLI_DATA (against keys storage and

		destruction)
T.TOE_PublicAuthKey_Forgery	NO	This threat is covered by the applet
T.Authentication_Replay	YES	This threat is mainly covered by the applet but the part addressed against the platform meets T.INTEG_APPLI_DATA (against storage and destruction of keys used for the authentications)