Security Target lite

Health Insurance Card G2

GEMALTO, B.P. 100, 13881 GEMENOS CEDEX, FRANCE.

Tel: +33 (0)4.42.36.50.00 Fax: +33 (0)4.42.36.50.90

Printed in France.

Document Reference: R0R23087_001_ASE_Lite V1.13

September 23, 2016

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. ST INTRODUCTION

1.1 ST REFERENCE

Security Target and associated evaluation are completely defined by information located in the following table.

Title:	ASE - Security Target Health Insurance Card G2
Reference:	R0R23087_001_ASE_Lite version : 1.13 date : 23/09/2016
Origin:	GEMALTO
ITSEF:	TUV Informationstechnik GmbH evaluation body.
Certification Body:	BSI
Evaluation scheme:	German

Table 1 –ST References

This Security Target describes:

- The Target of Evaluation, the TOE components, the components in the TOE environment, the product type, the TOE environment and life cycle, the limits of the TOE,
- The assets to be protected, the threats to be countered by the TOE itself during the usage of the TOE,
- The organizational security policies, and the assumptions,
- The security objectives for the TOE and its environment,
- The security functional requirements for the TOE and its IT environment,
- The TOE security assurance requirements,
- The security functions and associated rationales.

This ST has been built with the:
Common Criteria for Information Technology Security Evaluation Version 3.1, release 4, September 2012 which comprises [CCPART1], [CCPART2], and [CCPART3]

1.2 TOE Reference

Product and TOE are completely defined by information located in the following table.

Product Name	GeGKOS
Product Version	C1
TOE name	Health Insurance Card G2
TOE Version	1.0.0
Micro Controller	Infineon M7892 B11

Table 2 –TOE References

1.3 TOE overview

1.3.1 TOE type

The TOE " **Health Insurance Card G2 "** is a smart card IC with Embedded Software (the Card Operating System "GeGKOS C1") together with an external "wrapper" software for standardized interpretation of exported TSF data.
The actual applicative data structures and TSF data needed to build an initialized smart card product are explicitly excluded from TOE scope.

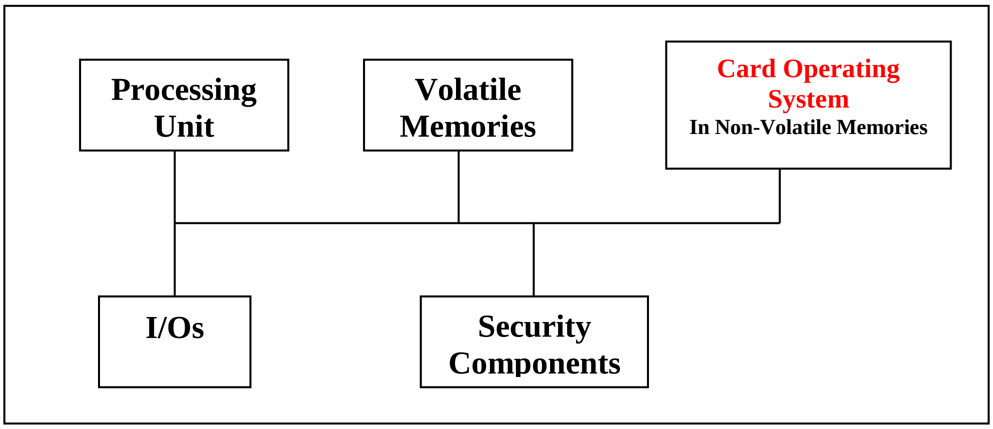


Figure 1 – Smart card IC with Embedded Software

The IC hardware comprises a processor, I/O modules, security circuits, volatile RAM, and NVM storing the COS (as part of the TOE).

The Smart Card Integrated circuit is the INFINEON M7892 B11 micro-controller. The evaluation of the **Health Insurance Card G2** is built on the results of the evaluation of the M7892 B11.

The TOE's Card Operation System is conformant to the gematik COS specification [gemSpec_COS], without supporting any of the normative options defined in chapter 2 of that document, namely "USB_Schnittstelle", "Kryptobox", "kontaktlose_Schnittstelle", "logische_Kanäle", and "PACE_PCD". By this choice it is tailored to instantiate an electronic Health Card (patient card), but no other type of card in the German Health IT infrastructure, like Health Professional Card or Secure Module Card.

The size of the card is type ID-1 according to ISO 7810 (the usual credit-card-size).

The card is a card with contacts according to ISO 7816-1 to –3.

2. INTENDED USE AND MAJOR SECURITY FEATURES

The TOE contributes to the Health application management by providing the following services:

- authentication of human user and external devices,
- storage of and access control on user data,
- key management and cryptographic functions,
- management of TSF data,
- life cycle support for TSF data, including secure production steps as specified in [GeGKOS_PERS].
- export of non-confidential TSF data of the object systems. Those data can be transformed into a standardized format by utilizing the Wrapper", an external software package that is part of the TOE but provides no security functionality.

The services mentioned are implemented with following cryptography:

- 3TDES, that is Triple DES using 192 bit symmetric keys.
- AES using 128, 192 and 256 bit keys.
- RSA with key size of 2048 bit and 3072 bit.
- hash computation with SHA-1, SHA-256, SHA-384 and SHA-512.
- Elliptic curve cryptography on following curves: ansix9p256r1, ansix9p384r1, brainpoolP256r1, brainpoolP384r1, and brainpoolP512r1.

To ensure the correct operation of the COS the TOE implements following security features:

- Storage of TOE data along with checksums to ensure integrity.
- Integrity and confidentiality of the embedded software (ES).
- TOE self protection by software design and utilization of the IC security features. For more details see § 8.13.

With the mechanisms above the TOE protects the assets described in section § **4.1.1** by fighting the following risks:

- Cloning: Substitution of programmed microchip (personalized or non-personalized Smart Card).
- Confidential data disclosure: Disclosure of confidential data in programmed microchip, i.e. Application code, keys, PINs.
- Non-integrity: Use of non-valid data.
- Identity usurpation: Management (i.e. personalization,) by unauthorized administrators. Use of Application by unauthorized user, i.e. other than the legitimate one.
- Physical attacks : the physical tampering of the TOE user data, TSF data or by modification of security features
- Information leakage : as emanations, variations in power consumption, I/O characteristics, clock frequency or by changes in processing requirements
- Malfunction due to an environment stress
- Use of functions in wrong phase to manipulate TOE’s security functions or features or TSF data

2.1 TOE Description

2.1.1 TOE definition

The TOE comprises the following parts:

TOE_IC, consisting of:

- the circuitry of the chip (the integrated circuit, IC) and
- the IC Dedicated Software with the parts IC Dedicated Test Software and IC Dedicated Support Software

TOE_ES,

- the Embedded Software, i.e. the operating system,

TOE_WRAPPER,

- the external application for interpretation of exported data,

and

guidance documentation delivered together with the TOE.

2.1.2 Global Description

In essence the TOE consists of the Gemalto implementation of the gematik COS specification (GeGKOS C1 operating system). It resides on the certified Infineon M7892 B11 contact interface micro-controller.

Also part of the TOE is an external software called “Wrapper”, used to translate exported non-confidential TSF data from the implementation specific format into a standardized one for use in an official verification tool. The Wrapper implementation complies with [GemSpec_COS-wrapper].

Therefore the TOE is a composed one, containing the following components for this composite evaluation:

Component	Supplier
Embedded Software and Wrapper (TOE_ES plus TOE_WRAPPER)	Gemalto
Micro-controller	Infineon

Table 3 - TOE components

The actual applicative data structures and TSF data needed to build an initialized smart card product are explicitly excluded from TOE scope.

2.1.3 Operating System Description

The GeGKOS C1 operating system (TOE_ES) meets the specification [gemSpec_COS].

The OS provides the following functions:

- a file system according to [ISO C4],
- access control for the file system and the cryptographic services,
- secure messaging for a secured communication with the external world,
- selection and management of security environments;
- user authentication with passwords,
- component authentication with symmetric and asymmetric cryptographic keys,
- import of external public keys via CVC verification,
- creation and verification of digital signatures, and
- enciphering and deciphering with asymmetric cryptography.

The data structures of the final product will determine the access to those functions and their execution modes by containing the appropriate access conditions and control information, e.g. key lengths or maximum PIN retry counters.

The TOE_ES consists of the following software modules:

The APDU Manager

This module implements the high level processing of all APDU commands supported.

The Access Manager

- accesses the module Object System to find the relevant access rules for the command to be executed and the data to be accessed. This provides full control over the TOE assets like applicative data, PINs, and keys.
- checks if authentication and Secure Messaging has occurred as requested by the access conditions.

The Access Enabling Mechanisms

This module includes:

- Authentication by human users and external components,
- Secure Messaging.

The Object System

All persistent data of the electronic health applications (including PINs and keys) are stored in the Object System. It manages all the key objects, PIN objects, access rule objects, DFs and EFs available.

The Cryptographic Computations

This is the package of cryptographic algorithms directly available at APDU processing or used for the Access Enabling Mechanisms.

A cryptographic library internally developed by GEMALTO supplies the basic cryptographic functionalities needed for these OS components, utilizing the chip's cryptographic co-processors:

- symmetric algorithms based on 3-DES (key size 24 bytes = 3 parts of 56 bits) or 16 bytes = 2 parts of 56 bits),
- symmetric algorithms based on AES (key sizes 128, 192, and 256 bits),
- asymmetric algorithms based on RSA (key size 2048, 3072 bits),
- asymmetric algorithms based on Elliptic Curves (ansix9p256r1, ansix9p384r1, brainpoolP256r1, brainpoolP384r1, and brainpoolP512r1),
- Hash algorithms: SHA-1 (only used in derivation of AES session keys), SHA-256, SHA-384, and SHA-512) ,

2.1.4 TOE security features

TOE implements following security features:

- All data in non-volatile memory (especially keys and PINs) are equipped with a checksum to detect integrity faults.
- The data structures of the card comprise hierarchical object system. EFs outside of the currently selected DF and other objects (PINs, keys, access rules) outside the current path are not accessible. Thus the object system forms a built-in way to establish data separation between different applications.
- After start-up, the integrity of filter code and other critical TSF data is verified. All authentication states are deleted.
- Self protection is achieved by software design features such as checking hardware registers, desynchronization, redundancy, usage of platform‘s protection like clock jitter, and self test features environmental sensors. Sensitive data in NVM are masked.
- In case an object in NVM is deleted, the associated memory area is cleared.

2.1.5 Hardware Platform

The TOE contains software and hardware identified as Infineon M7892 B11 and certified by BSI with the certificate reference BSI-DSZ-CC-0782-V2-2015
(Confirmation of the reassessment - dated from 05/09/2013). The IC is compliant with the [BSI-CC-PP-0035-2007]. The IC is certified at the level EAL6augmented with ALC_DVS.2, AVA_VAN.5 components.
The Infineon M7892 provides algorithms to the embedded software Rivest-Shamir-Adleman Cryptography (RSA), Elliptic Curve (EC) and Secure Hash Algorithm (SHA-256), Advanced Encryption Standard (AES) and Triple Des Encryption Standard (3DES) but they are not used by the composite TOE. These algorithms are developed by Gemalto using hardware accelerators for cryptographic computation and are part of the embedded software.

This certified IC is described in the platform's Security Target [ST IC].
Besides state of the art attack resistance this IC provides a physical (PTG2) random number generator. The TOE_ES uses it for its cryptographic computations.

2.1.6 TOE Boundaries

The following figures illustrate the TOE physical and logical boundaries.

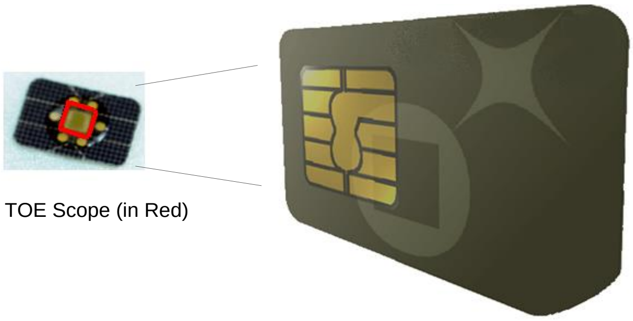


Figure 2 – TOE Physical Boundaries

Figure 2 shows the TOE as a smartcard including a plastic card body and a module performing the interface between reader and the embedded chip. The Target of Evaluation (TOE) is the Smart Card Integrated Circuit with Embedded Software (without applicative data structures) in accordance to its functional specifications. The physical boundaries are then the contacts and the surface of the chip module.
Other smart card product items (such as plastic, module, security printing…) are outside the scope of this evaluation.
The external wrapper is pure software and has no physical scope. So it is omitted in this figure.

Figure 3 below illustrates the logical boundaries of the smart card part of the TOE.

The physical scope is framed by the grey line.
The logical scope is highlighted in yellow: it is the chip with all the embedded software but without applicative data structures. The wrapper is inside the logical TOE boundary, too.

All the ES modules are included inside the TOE (see the ***TOE enforcing element***). This software uses the hardware and its firmware to provide the TOE functionality. The hardware and its firmware is part of the TOE.

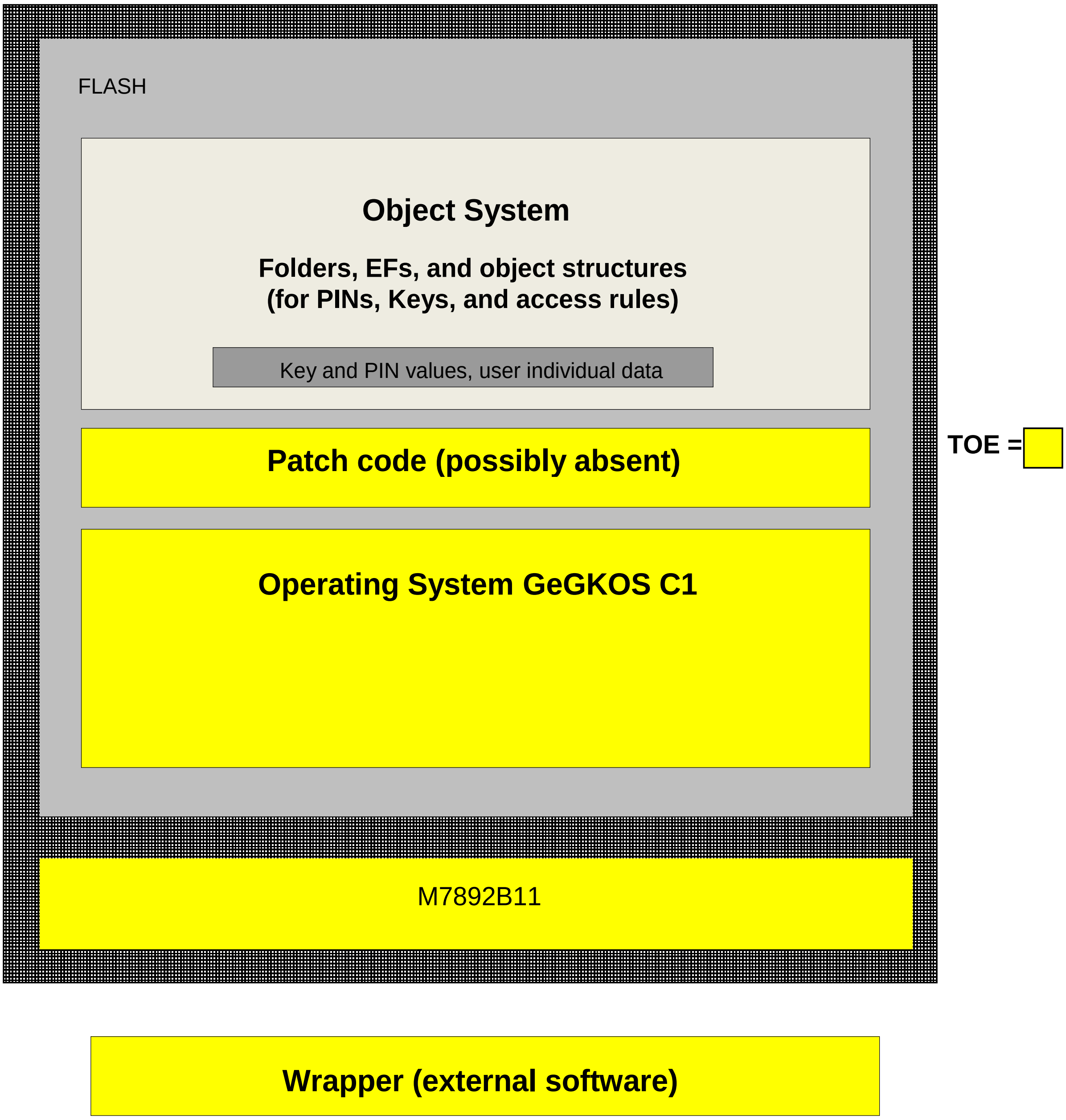

FLASH
Object System
Folders, EFs, and object structures
(for PINs, Keys, and access rules)
Key and PIN values, user individual data
Patch code (possibly absent)
Operating System GeGKOS C1
M7892B11
TOE =
Wrapper (external software)

Figure 3 – TOE logical boundaries

2.1.7 TOE Life Cycle and TOE Actors

A Smart Card's life cycle is decomposed in several phases.

Each life cycle phase is linked to certain TOE actors. This is shown in table below. Further details of the single phases follow in the subsections below.

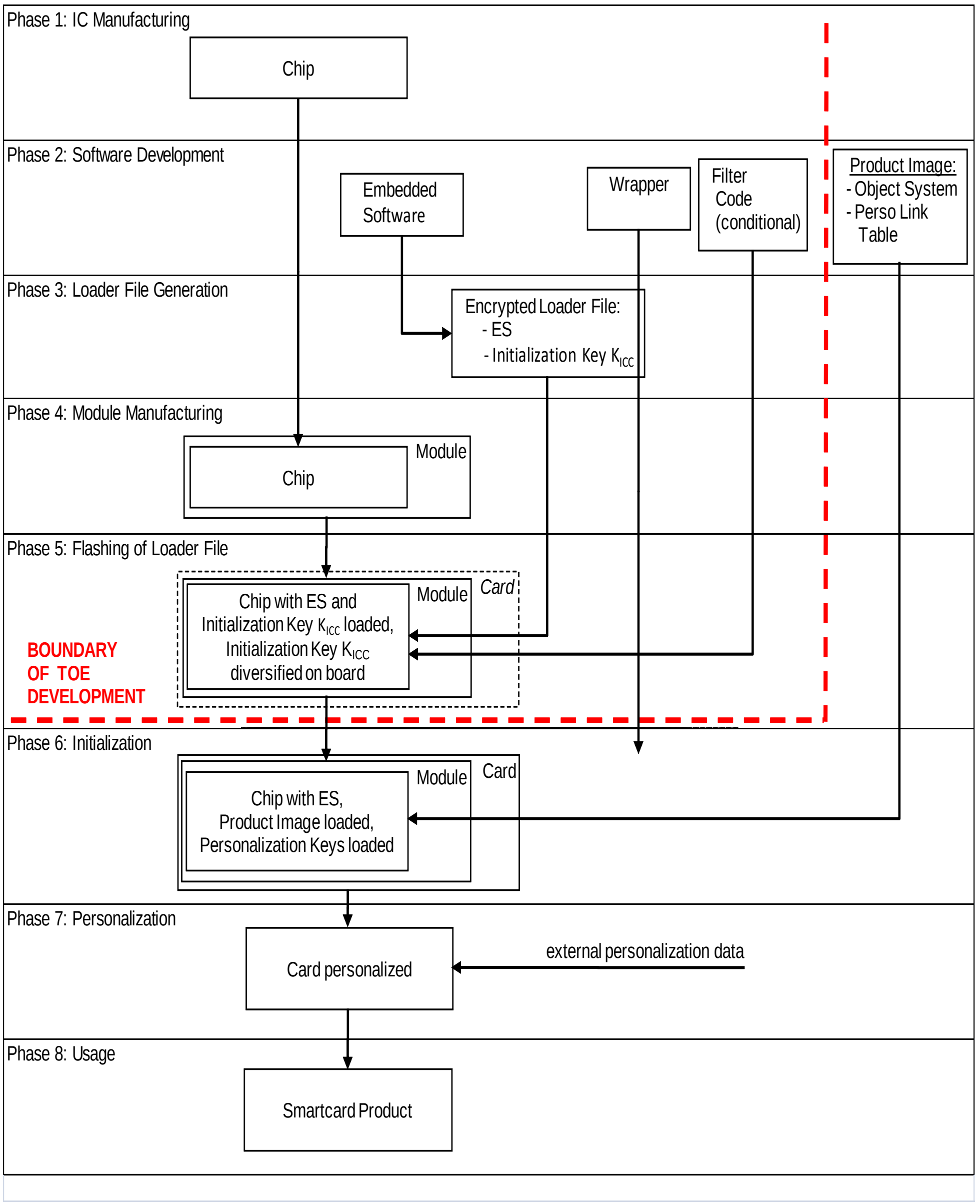

Phase 1: IC Manufacturing
Chip
Phase 2: Software Development
Embedded Software
Wrapper
Filter Code (conditional)
Product Image:
- Object System
- Perso Link Table
Phase 3: Loader File Generation
Encrypted Loader File:
- ES
- Initialization Key K_ICC
Phase 4: Module Manufacturing
Chip
Module
Phase 5: Flashing of Loader File
Chip with ES and Initialization Key K_ICC loaded, Initialization Key K_ICC diversified on board
Module
Card
BOUNDARY OF TOE DEVELOPMENT
Phase 6: Initialization
Chip with ES, Product Image loaded, Personalization Keys loaded
Module
Card
Phase 7: Personalization
Card personalized
external personalization data
Phase 8: Usage
Smartcard Product

Figure 4 –Health Insurance Card G2 Lifecycle

The following table presents the TOE actors, and logical phase associated with each step of the life cycle

Phase	TOE phase	Industrial deliverable	TOE actors
1 2	IC Manufacturing Software Development	Wafers with ICs ES code and Product Image	IC manufacturer Product developer
3	Loader File Generation	Loader File	Product developer
4	Module Manufacturing	Modules	Module manufacturer
5	Flashing of Loader File	Modules or Card pre-initialized	Card manufacturer
6	Initialization	Card initialized	Initializer
7	Personalization	Card personalized	Personalizer
8	Usage	Smartcard	TOE end users: - Card holder, - devices of the Health IT infrastructure, among others representing the card issuer

Table 4 - TOE life cycle

2.1.7.1 Phase 1: IC Manufacturing

The IC Manufacturer is in charge of producing the IC and the test operation. This phase is covered by the IC evaluation.

For this product the IC manufacturer is **INFINEON**.

2.1.7.2 Phase 2: Software Development

The ES is developed in phase 2 by the development team of the Product developer. This team also generates the Product Image, which comprises the data structures for the Object System, a table defining the data to be personalized and linking them to their target locations in NVM. This team also generates a Filter Code if needed. Note that the Filter Code is part of the TOE, while the other data structures are not.
Only members of the development team are able to generate such a product image and the filter code.

The Product Image is delivered to the Initializer for initialization in phase 6. During initialization only dedicated commands are allowed, and the TOE accepts authentic images only.

The filter code is delivered to Card Manufacturer for loading in phase 5.

For this product, the Product developer is **GEMALTO**.

2.1.7.3 Phase 3: Loader File Generation

After completion of the ES, the Product developer submits it to an internal trust-center that generates an encrypted Loader File from it, which also contains a master key to secure the initialization step.

The Loader File is delivered to the Card Manufacturer for pre-initialization.

For this product, the Product developer is **GEMALTO.**

2.1.7.4 Phase 4: Module Manufacturing

The Module manufacturer is responsible for manufacturing modules from the ICs provided by the IC manufacturer.

For this product, the Module Manufacturer is **GEMALTO.**

2.1.7.5 Phase 5: Flashing of Loader File

The presentation of the flash loader key is necessary for access. The ES and the master key for the initialization phase are flashed onto the chip.
After that the ES is started for the first time, and the master key for initialization is replaced by diversifying it on-board with chip individual data.
Possibly the Filter code is loaded during this phase.

The Card Manufacturer also loads keys to secure the subsequent initialization and personalization steps.

By completion of this production step the TOE comes into existence.

During this phase the module could already be embedded into a smart card.

For this product, the Card Manufacturer is **GEMALTO.**

2.1.7.6 Phase 6: Initialization

The presentation of a card individual authentication key is necessary for access. The Product Image is loaded onto the chip: Object System and personalization link table.

The Initialization phase can only be completed when the TOE has successfully checked the authenticity of the loaded Product Image.

In case the module embedding wasn’t performed during the flashing phase, the modules are now embedded into smart cards.

For this product, the Initializer is **GEMALTO** and is the same entity that performs also the preceding production step (phase 5). In fact the phases 5 and 6 are joined into one single production flow, which takes place without interruption. So the Card Manufacturer and the Initializer are the same entity, playing two roles.

2.1.7.7 Phase 7: Personalization

The encrypted personal data (i.e. cards specific keys, card holder data, …) are transferred to and decrypted inside the IC. Unique Keys are used for the encryption of each personalization data record.

For this product, the Personalizer is **GEMALTO or SWISS POST SOLUTIONS (SPS).**

2.1.7.8 Phase 8: Usage

The Card Issuer is responsible during **Phase 8** for the smartcard product delivery to the smartcard end-user (the card holder), and the end of life process.

The authorized personalization agents (card management systems) might be allowed to add new applications and to modify or delete existing applications. But it is not possible to load additional executable code. The access to those specifically secured functions for this usage phase are mediated by the access conditions set up with the object system (for example they require card-to-card authentication and secure messaging). This functionality doesn't mean that the card can be switched back to an earlier life cycle stage.

The card holder in the operational usage phase uses this TOE as a patient card in the eHealth IT infrastructure..

For this product, the Card Issuer is a health insurance.

2.1.8 TOE delivery

The TOE is formally delivered after phase 5 "Flashing of Loader File" as an IC already embedded in the plastic card and containing and the ES, but no data structures.

The TOE will be delivered as:

(1) Documentation:

- Administrator Guidance for the COS [PRE_GUIDE] :

	Reference	**Title**	**Version**	**Date**
PRE_GUIDE	R0R23087_001_PRE	Preparative Procedures	1.16	23/09/2016

- User Guidance for COS [OPE_GUIDE] and Wrapper [AGD_Wrapper] :

	Reference	**Title**	**Version**	**Date**
OPE_GUIDE	R0S23087_001_OPE_ES	Operational User Guidance ES Electronic Health Card	1.23	13/09/2016
AGD_Wrapper	R0S23087_001_AGD Wrapper	User Guidance for Wrapper, Electronic Health Card - GEGKOS C1	1.8	22/09/2016

- Additional user documentation, referenced by the above Guidances:
  Note that these are only delivered to gemalto internal entities.

	Reference	**Title**	**Version**	**Date**
GEGKOS_SRS	R0S23087_001_SRS	Software Requirements Specification For GeGKOS C1	B12	22/09/2016
CIS	R0S23087_CIS	Card Initialization Specification GEGKOS C1	A05	22/09/2016
GEGKOS_PERS	R0S23087_001_ GeGKOS_Perso	Perso Manual for GEGKOS C1	1.12	22/09/2016

(2) HW-Part of TOE:

- Chip modules with Infineon M7892 B11 embedded into smart cards,.

(3) SW Part of the TOE*:

- The ES loaded onto the chip modules, and
- The external Wrapper as JAVA archive.
- Kicc: Initialisation key[1]
- K_Verify:
- Perso Keys (Kenc, Kmac, Kdec)

(4) Keys for card production (outside the card)

- MK_ICC
- KMC

Note*: The keys mentioned shall be understood as on-card key containers. The key values are out of TOE scope.

[1] The MK_ICC is only formally delivered by the Card-Manufacturer to the Initializer. The Card Manufacturer already has the MK_ICC available, and directly following phase 5 he also takes the role of the Initializer. There is no separate entity performing the initialization as individual step, so no physical delivery of MK_ICC takes place.

2.1.9 TOE actors summary

The TOE actors as mentioned in the subsection of TOE Life Cycle are summarized in the following, categorized as "Producers", covering all actors until TOE completion, or "TOE users" in the strict sense of the CC formalism.

TOE producers

The TOE producers are as follows:

Producers	**Description**
Product Developer	The Product developer designs the IC ES. For this product, the developer is **GEMALTO (phase 2)**.
IC Manufacturer	The IC manufacturer -or founder- designs, manufactures, and loads the ES in the IC. For this product, the IC manufacturer is **INFINEON (phase 1)**.
Module Manufacturer	The module manufacturer processes the ICs to modules. For this product, the module manufacturer is **GEMALTO (phase 4)**.
Card Manufacturer	The Card manufacturer is responsible: • For embedding the modules provided by the module manufacturer into Smart Cards (phase 5) • for pre-initialization of the Smart Cards (loading the ES and a secret key for the initialization phase). For this product, the Card manufacturer is **GEMALTO.**

Table 5 – TOE Producers list (in the scope of the TOE)

Initializer	The Initializer loads the Product image onto the card and establishes keys for the personalization phase. For this product, the Initializer is **GEMALTO** (phase 6).[Card Manufacturer and Initializer is the same entity]
Personalizer	The Personalizer processes the card individual personalization data and loads them onto the card (phase 7). For this product, the Personalizer is GEMALTO or SWISS POST SOLUTIONS (SPS).
Card issuer	The Card issuer -short named « issuer » issues cards to its customers that are the « End users ». The card belongs to the Card issuer. Therefore, the Card Issuer is responsible for: • Providing personalization data to the Personalizer • Distribution of the cards. • Maintenance of the cards (i.e. unblocking the PIN) • Invalidation of the cards. For this product, the Card Issuer are Health insurance agencies (phase 8).

Table 6 –Producers list (out of the scope of theTOE)

TOE users

The TOE users are listed below. This definition adopts the users already defined in the PP.

Users	Description
Initializer	The Initializer is responsible for initializing the card with the Product image and establishing the authentication keys for personalization phase.[Card manufacturer and Initializer is the same entity]
Personalizer	The Personalizer personalizes the card by loading the Card issuer and End user data as well as Application secrets such as cryptographic keys and PIN values. The personalization includes printing of the card holder specific visual readable data onto the physical smart card.
Human user	This is the cardholder of the usage phase, as customer of a health insurance (card issuer). The card is personalized with the card holder's identification and secrets
Device	This means any device of the eHealth IT infrastructure, communicating with the TOE in usage phase via terminals, e.g. Card Management Systems representing the health insurance as card issuer, or Health professional cards.

Table 7 –Users list

2.1.10 Non-TOE hardware/software/firmware

In principle the smart card part of the TOE, comprising chip and ES, is a standalone device without any dependency on non-TOE elements. However, for communication with the 'external world' it would need a terminal (card reader) with ISO contacts.

The Wrapper, being pure software delivered as a java archive (*.jar), needs to be executed on a machine running Java Standard Edition Runtime Environment 7 or higher.

3. CONFORMANCE CLAIMS

3.1 CC CONFORMANCE CLAIMS

This security target claims to be conformant to the Common Criteria version 3.1, which comprises of:

- Common Criteria for Information Technology Security Evaluation (CC), V3.1, Part 1: Introduction and general model, Revision 4, September 2012[CCPART1].
- Common Criteria for Information Technology Security Evaluation (CC), V3.1, Part 2: Security functional components, Revision 4, September 2012 [CCPART2].
- Common Criteria for Information Technology Security Evaluation (CC), V3.1, Part 3: Security assurance components, Revision 4, September 2012 [CCPART3].

as follows:

- Part 2 extended
- Part 3 conformant

- Common Methodology for Information Technology Security Evaluation [CEM], V3.1, Revision 4, September 2012,

The evaluation is performed according [CEM] and supporting documents [AIS 36].

3.2 PP CLAIM

This ST claims strict conformance to [PP eHC] .

The TOE includes an Integrated Circuit certified with CC EAL6 augmented with ALC_DVS.2 and AVA_VAN.5.

3.3 PACKAGE CLAIMS

This ST is conformant to the EAL4 package as defined in [CCPART3].

None of the packages linked to the implementation options of [gemSpec_COS] are included.

The assurance level is EAL4 augmented with:

- AVA_VAN.5 Advanced methodical vulnerability analysis
- ALC_DVS.2 Sufficiency of security measures
- ATE_DPT.2 Testing : security enforcing modules

3.4 CONFORMANCE RATIONALE

This ST is claimed to be conformant to the above mentioned PP [PP eHC]. A detailed justification is given in the following by

- describing some single aspects which are main issues of PP conformance, and
- describing differences between the ST and the PP.

3.4.1 Main aspects

- Text from introduction, TOE overview, TOE description has been taken from the PP [PP eHC, §1.2] and specific informations linked to the product have been added
- All definitions of the security problem definition in [PP eHC, §3] have been included in the ST in the same wording.
- All definitions of the security objectives in [PP eHC, §4] have been included exactly in the same wording as the PP.
- The SFR defined in the extended components definition of [PP eHC, §5] has been included in the ST exactly in the same wording as the PP.
- All SFRs for the TOE from the [PP eHC, §5] have been included in the ST exactly in the same wording as the PP and filling all necessary selections or assignments.The security assurance requirements (SARs) are originally taken from SARs of CC 3.1 Part 3 according to the package conformance EAL 4 augmented with AVA_VAN.5, ALC_DVS.2 and ATE_DPT.2 .
- The structure of the ST is taken from the PP added by the section 7 (TOE summary specification)

3.4.2 Differences between ST and PP

1) The ST adds the following assets to those of the PP :
   - Product Image
   - Key “KICC” to secure the initialization step.
   - Key “K_Verify” to secure the Product image
   - “Perso Keys” to secure the personalization step.
2) The ST adds the following external entities to those of the PP :
   - Initializer
   - Personalizer
3) The ST suppresses the reference to PACE protocol in operations and security attributes
   - Table 28 : suppress for Security attribute SesionKeyContext element negotiationKeyInformation, for security attribute globalSecuritylist element CHA or KeyIdentifier,
   - Table 29 : suppress for commands PSO Decipher, General Authenticate suppress SFR linked to PACE, and
   - Table 35 : suppress FCS_CKM.1/ DH.PACE as SFR and as dependency for FCS.CKM.4.
4) The ST suppresses the word ‘Shareable’ as the TOE do not implement it because of option linked to the logical channel package (table 28)
5) The ST creates in Subjects, objects, operations and security attributes table the Rule object
6) The optional commands CREATE, ENVELOPE, GET RESPONSE, PSO HASH, SEARCH BINATY, WRITE RECORD are not implemented
7) The commands PSO COMPUTE/VERIFY CRYPTOGRAPHIC CHECHSUM are not implemented as option crypto box is not supported.
8) The ST updates the FDP_SDI SFR:
   FDP_SDI is iterated with varying operation: FDP_SDI.2/ReadEF and FDP_SDI.2/Internal
9) The ST introduces additional iterations FMT_MTD.1/Init and FMT_MTD.1/Perso to cover authentication need for Initialization and Personalization process. FMT_SMR.1.1 is amended accordingly.

10) FCS_COP.1/PAUTH is added for authentications in production process, together with the dependencies chain FDP_ITC.1/PKEYS, FDP_ACC.1/PKEYS, and FDP_ACF.1/PKEYS. FMT_MSA.3 was amended accordingly.
11) The ST enhances FCS_COP.1.1/COS.RSA and FCS_COP.1.1/COS.ELC to support additional RSA modulo length of 3072 bit for RSA public key operation. Those keys are not stored in the card's nvm but are taken from the command data of the PSO ENCIPHER and PSO TRANSCIPHER commands.
12) The ST introduces more precision concerning initialization and personatisation commands in FMT_SMF.1.1(points 1 and 2)

4. SECURITY PROBLEM DEFINITION

4.1 General

The Security Problem Definition (SPD) is the part of the ST, which describes :

- Assets the TOE shall protect.
- Subjects, who are users (human or system) of the TOE or who might be threats agents (i.e. attack the security of the assets).
- Operational security policies, which describe overall security requirements defined by the organization in charge of the overall system including the TOE (in particular this may include legal regulations, standards and technical specifications).
- Threats against the assets, which shall be averted by the TOE together with its environment.
- Assumptions on security relevant properties and behavior of the TOE's environment.

4.1.1 Assets

User data in EF	Data for the user stored in elementary files of the file hierarchy.
Secret keys	Symmetric cryptographic key generated as result of mutual authentication and used for encryption and decryption of user data.
Private keys	Confidential asymmetric cryptographic key of the user used for decryption and computation of digital signature.
Public keys	Integrity protected public asymmetric cryptographic key of the user used for encryption and verification of digital signatures and permanently stored on the TOE or provided to the TOE as parameter of the command.
Product Image	Initialization data, mainly data structures for the Object System. In addition it contains the secret authentication keys for the personalizer and the table of links that define where data personalized would be filled into the object system.
Key « Kicc »	Card individual key to authenticate before initialization (derive from card serial number)
« K_Verify »	Key used by the developer to secure the Product Image, and by the card itself to check the Product Image.
« Perso Keys »	Card individual keys to secure the personalization

Table 8 – Data Objects list

Note: elementary files (EF) are stored in the MF, any DF, Application or Application Dedicated File. The place of an EF in the file hierarchy defines features of the User Data stored in the EF. User data does not affect the operation of the TSF (cf. CC part 1, para. 100). Cryptographic keys used by the TSF to verify authentication attempts of external entities (i.e. authentication reference data) including the verification of Card Verifiable Certificates (CVC) or authenticate itself to external entities by generation of authentication verification data in a cryptographic protocol are TSF data.

4.1.2 External entities

World	Any user independent on identification or successful authentication
Human User	A person authenticated by password or PUC.
Device	An external device authenticated by cryptographic operation
Initializer	An entity responsible for initializing the card with the Product Image
Personalizer	An entity who loads the card individual data of Card issuer and End user

Table 9 – External entities / Subjects list

4.2 THREATS

This section describes the threats to be averted by the TOE independently or in collaboration with its IT environment. These threats result from the assets protected by the TOE and the method of TOE's use in the operational environment.

IC threat label	IC threat title	IC threat content
T.Leak-Inherent	Inherent Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets. No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements.
T.Phys-Probing	Physical Probing	An attacker may perform physical probing of the TOE in order (i) to disclose User Data (ii) to disclose/reconstruct the Security IC Embedded Software or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded
T.Malfunction	Malfunction due to Environmental Stress	An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions.
T.Phys-Manipulation	Physical Manipulation	An attacker may physically modify the Security IC in order to (i) modify User Data (ii) modify the Security IC Embedded Software

IC threat label	IC threat title	IC threat content
		(iii) modify or deactivate security services of the TOE, or (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded
T.Leak-Forced	Forced Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets even if the information leakage is not inherent but caused by the attacker.
T.Abuse-Func	Abuse of Functionality	An attacker may use functions of the TOE which may not be used after TOE Delivery in order to (i) disclose or manipulate User Data (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software.
T.RND	Deficiency of Random Numbers	An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.

Table 10 –Threats for the IC and taken over into this ST

The threats are those defined by the eHC PP.

T.Forge_Internal_Data	**Forge of User or TSF data :** An attacker with high attack potential tries to forge internal user data or TSF data. This threat comprises several attack scenarios of smart card forgery. The attacker may try to alter the user data e.g. to add user data in elementary files. The attacker may misuse the TSF management function to change the user authentication data to a known value.
T.Compromise_Internal_Data	**Compromise of confidential User or TSF data :** An attacker with high attack potential tries to compromise confidential user data or TSF data through the communication interface of the TOE. This threat comprises several attack scenarios e.g. guessing of the user authentication data (password) or reconstruction the private decipher key using the response code for chosen cipher texts (like Bleichenbacher attack for the SSL protocol implementation), e.g. to add keys for decipherment. The attacker may misuse the TSF management function to change the user authentication data to a known value.
T.Misuse	**Misuse of TOE functions :** An attacker with high attack potential tries to use the TOE functions to gain access to the access control protected assets without knowledge of user authentication data or any implicit authorization. This threat comprises several attack scenarios e.g. the attacker may try circumvent the user authentication to use signing functionality without authorization. The attacker may try to alter the TSF data e.g. to extend the user rights after successful authentication.
T.Malicious_Application	**Malicious Application :** An attacker with high attack potential tries to use the TOE functions to install an additional malicious application in order to compromise or alter User Data or TSF data.
T.Crypto	**Cryptographic attack against the implementation:** An attacker with high attack potential tries to launch a cryptographic attack against the implementation of the cryptographic algorithms or tries to guess keys using a brute-force attack on the function inputs. This threat comprises several attack scenarios e.g. an attacker may try to foresee the output of a random number generator in order to get a session key. An attacker may try to use leakage during cryptographic operation in order to use SPA, DPA, DFA or EMA techniques in order to compromise the keys or to get knowledge of other sensitive TSF or User data.

	Furthermore an attacker could try guessing the key by using a brute-force attack.
T.Intercept	**Interception of Communication :** An attacker with high attack potential tries to intercept the communication between the TOE and an external entity, to forge, to delete or to add other data to the transmitted sensitive data. This threat comprises several attack scenarios. An attacker may try to read or forge data during transmission in order to add data to a record or to gain access to authentication data.
T.WrongRights	**Wrong Access Rights for User Data or TSF Data:** An attacker with high attack potential executes undocumented or inappropriate access rights defined in object system and compromises or manipulate sensitive User data or TSF data

Table 11 – Threats list

4.2.1 Assets coverage

The following table provide an overview of how the threats are relevant for the assets of the TOE

Threats / Assets	data in EF	Secret keys	Private Keys	Public keys	Product Image	Kicc	K_Verify	Perso Keys
T.Forge_Internal_Data	**X**			**X**	**X**	**X**	**X**	**X**
T.Compromise_Internal_Data	**X**	**X**	**X**			**X**	**X**	**X**
T.Misuse	**X**							
T.Malicious_Application	**X**							
T.Crypto		**X**				**X**	**X**	**X**
T.Intercept	**X**			**X**				
T.WrongRights	**X**							

Table 12 – Threats / Assets correspondence analysis

4.3 ORGANISATIONAL SECURITY POLICIES

The TOE and/or its environment shall comply with the following Organisational Security Policies (OSP) as security rules, procedures, practices, or guidelines imposed by an organisation upon its operation.

IC OSP label	**IC OSP content**	**Link to the composite product**
P.Process-TOE	Protection during TOE Development and Production: An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.	No contradiction with the present evaluation; the chip traceability information is used to identify the composite TOE.

Table 13 – Organisational Security Policies

4.4 Assumptions

IC assumption label	IC assumption title	IC assumption content	Link to the composite product
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation	While the TOE is delivered after Phase 3 Loader file generation and Phase 4 Module manufacturing the current TOE is delivered after Phase 5 Flashing of loader file before Phase 6 Personalisation. The protection during Phase 4 and during Phase 5 is addressed by security of the development environment of the current TOE. Only protection during Personalisation is in responsibility of the operational environment. So it is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).	A.Process-Sec-SC
A.Plat-Appl	Usage of Hardware Platform	Usage of Hardware Platform as TOE as addressed by A.Plat-Appl is covered by ADV class related to COS as part of the current TOE.	Removed
A.Resp-Appl	Treatment of User Data	All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.	A.Resp-ObjS

Table 14 – Composition – Assumptions part

A.Process-Sec-SC	**Protection during Initialization and Personalisation** It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).
A.Plat-COS	**Usage of COS** An object system designed for the TOE meets the following documents: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the user guidance, and the application notes, and (ii) findings of the TOE evaluation reports relevant for the COS as documented in the certification report.
A.Resp-ObjS	**Treatment of User Data by the Object System** All User Data and TSF Data of the TOE are treated in the object system as defined for its specific application context.

Table 15 – Assumptions list

5. SECURITY OBJECTIVES

5.1 GENERAL

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

5.2 SECURITY OBJECTIVES FOR THE TOE

IC TOE security objective Label	**IC TOE security objective Title**
O.Leak-Inherent	Protection against Inherent Information Leakage
O.Phys-Probing	Protection against Physical Probing
O.Malfunction	Protection against Malfunctions
O.Phys-Manipulation	Protection against Physical Manipulation
O.Leak-Forced	Protection against Forced Information Leakage
O.Abuse-Func	Protection against Abuse of Functionality
O.Identification	TOE Identification
O.RND	Random Numbers

Table 16 – Security Objectives for the IC and taken over into this ST.

O.Integrity	**Integrity of internal data** The TOE must ensure the integrity of the User Data, the security services and the TSF data under the TSF scope of control.
O.Confidentiality	**Confidentiality of internal data** The TOE must ensure the confidentiality of private keys and other confidential User Data and confidential TSF data especially the authentication data, under the TSF scope of control against attacks with high attack potential.
O.Resp-COS	**Treatment of User and TSF Data** The User Data and TSF data (especially cryptographic keys) are treated by the COS as defined by the TSF data of the object system.
O.TSFDataExport	**Support of TSF data export** The TOE must provide correct export of TSF data of the object system excluding confidential TSF data for external review.
O.Authentication	**Authentication of external entities** The TOE supports the authentication of human users and external devices. The TOE is able to authenticate itself to external entities.
O.AccessControl	**Access Control for Objects** The TOE must enforce that only authenticated entities with sufficient access control rights can access restricted objects and services. The access control policy of the TOE must bind the access control right of an object to authenticated entities. The TOE must provide management functionality for access control rights of objects.
O.KeyManagement	**Generation and import of keys** The TOE must enforce the secure generation, import, distribution, access control and destruction of cryptographic keys. The TOE must support the public key import from and export to a public key infrastructure.
O.Crypto	**Cryptographic functions** The TOE must provide cryptographic services by implementation of secure cryptographic algorithms for hashing, key generation, data confidentiality by symmetric and asymmetric encryption and decryption, data integrity protection by symmetric MAC and asymmetric signature algorithms, and cryptographic protocols for symmetric and asymmetric entity authentication.
O.SecureMessaging	**Secure messaging** The TOE supports secure messaging for protection of the confidentiality and the integrity of the commands received from successful authenticated device and sending responses to this device on demand of the external application. The TOE enforces the use of secure messaging for receiving commands if defined by access condition of an object.

Table 17 – TOE's objectives list

5.3 Security Objectives for the Operational Environment

The security objectives for the operational environment defined in the IC security target [ST IC] are addressed in new security objectives for the operational environment of the current TOE. The table below lists and maps these security objectives for the operational environment with the corresponding reference.

Security objective for the operational environnement define in IC Secrity target	Security objective for the operational environnement title define in IC security target	Rationale of the changes	Refined security objectives for the operational environment of the current TOE
OE.Plat-Appl	Usage of Hardware Platform	OE.Plat-Appl requires the Security IC Embedded Software to meet the guidance documents of the Security IC. The Security IC Embedded Software is part of the current TOE. This requirement shall be fulfilled in the development process of the TOE.	removed
OE.Resp-Appl	Treatment of User Data	OE.Resp-Appl requires the Security IC Embedded Software to treat the user data as required by the security needs of the specific application context. This objective shall be ensured by the TOE and the object system.	OE.Resp-ObjS
OE.Process-Sec-IC	Protection during composite product manufacturing	The policy defined for the Security platform IC is extended to the current TOE.	OE.Process-Card

Table 18 – Composition – Security objectives for the environment part

OE.Plat-COS	**Usage of COS** To ensure that the TOE is used in a secure manner the object system shall be designed such that the requirements from the following documents are met: (i) user guidance of the COS, (ii) application notes for the COS (iii) other guidance documents, and (iv) findings of the TOE evaluation reports relevant for applications developed for COS as referenced in the certification report.
OE.Resp-ObjS	**Treatment of User Data** All User Data and TSF Data of the object system are defined as required by the security needs of the specific application context.
OE.Process-Card	**Protection of Smartcard during Initialization and Personalisation** Security procedures shall be used after delivery of the TOE during Phase 6 Smartcard initialization and phase 7 personalisation up to the delivery of the smartcard to the end-user in order to maintain confidentiality and integrity of the TOE and to prevent any theft, unauthorised personalization or unauthorised use.

Table 19 – Environment's objectives list for the Electronic Health Application

5.4 SECURITY OBJECTIVES RATIONALE

5.4.1 Security Objectives Coverage and sufficiency

The following tables provide an overview for the coverage of the defined security problem by the security objectives for the TOE and its environment. The tables are addressing the security problem definition as given in the BSI-CC-PP-0035-2007 and the additional threats, organizational policies and assumptions defined in the current ST. It shows that all threats and OSPs are addressed by the security objectives for the TOE and for the TOE environment. It also shows that all assumptions are addressed by the security objectives for the TOE environment.

	(SAR ALC for IC part of the TOE)	OE.Process-Sec-Card	(SAR ADV class for COS part of the TOE)	(SAR for COS part of the TOE)	OE.Resp-ObjS	O.Identification	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND
A.Process-Sec-IC	(X)	(X)											
A.Process-Sec-SC		X											
A.Plat-Appl			(X)										

A.Resp-Appl				(X)									
A.Resp-ObjS					X								
P.Process-TOE						X							
T.Leak-Inherent							X						
T.Phys-Probing								X					
T.Malfunction									X				
T.Phys-Manipulation										X			
T.Leak-Forced											X		
T.Abuse-Func												X	
T.RND													X

Table 20 – Security Objective Rationale related to the IC platform

The **A.Process-Sec-IC** assumes and **OE. Process-Sec-IC** requires that security procedures are used after delivery of the IC by the IC Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). Development and production of the Security IC is part of development and production of the TOE because it includes the Security IC. The **A.Process-Sec-SC** assumes and **OE.Process-Sec-Card** requires security procedures from Phase 6 Smartcard initialization up to the delivery of the smartcard to the end-user. More precisely, the smartcard life cycle according to CCDB-2010-03-001 (cf. also to BSI-PP- 0035) are covered as follows.

- IC development (Phase 2) and IC manufacturing and testing (Phase3) are covered as development and manufacturing of the security ICand therefore of the TOE as well.

- IC packaging and testing (Phase 3) may be part of the development and manufacturing environment or the operational environment of the security IC. Even if it is part of the operational environment of the Security IC addressed by OE. Process-Sec-IC it will be part of the development and manufacturing environment of the current TOE and covered by the SAR ALC_DVS.2.

- IC packaging and testing (Phase 4) and Smartcard Packaging and finishing process (Phase 5) are addressed by OE. Process-Sec-IC but they are part of the development and manufacturing environment of the current TOE and covered by the SAR ALC_DVS.2.

- Smartcard initialization (phase 6), personalisation (phase 7) up to the delivery of the smartcard to the end-user is addressed by A.Process-Sec-IC and A.Process-Sec-SC and covered by OE.Process-Sec-Card.

The assumption **A.Plat-Appl** assumes that the Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report. This is met by the SAR of ADV class and the requirements for composite evaluation CCDB-2007-09-001.

The assumption **A.Resp-Appl** assumes that security relevant user data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context. This assumption is split into requirements for the COS part of the TSF to provide appropriate security functionality for the specific application context as defined by SFR of the current ST and the assumption **A.Resp-ObjS** that assumes all User Data and TSF Data of the TOE are treated in the object system as defined for its specific application context. The security objective for the operational environment **OE.Resp-Obj** requires the object system to be defined as required by the security needs of the specific application context.

The **OSP P.Process-TOE** and the threats **T.Leak-Inherent**, **T.Phys-Probing**, **T.Malfunction**, **T.Phys-Manipulation**, **T.Leak-Forced**, **T.Abuse-Func** and **T.RND** are covered by the security objectives as described in BSI-CC-PP-0035-2007. As stated in section 2.4, this ST claims conformance to BSI-CC-PP-0035-2007. The objectives, assumptions, policies and threats as used in Table 4 are defined and handled in BSI-CC-PP-0035-2007. Hence, the rationale for these items and their correlation with Table 4 is given in BSI-CC-PP-0035-2007and not repeated here.

The current ST defines new threats and assumptions for the TOE extended to the the Security platform IC as TOE defined in BSI-CC-PP-0035-2007 and extends the policy P.Process-TOE to the current TOE.

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControm	O.KeyManagement	O.Crypto	O.SecureMessaging	OE.Plat-COS	OE.Resp-ObjS	OE.Process-Card
T.Forge_Internal_Data	X		X									
T.Compromise_Internal_Data		X	X				X					
T.Malicious_Application				X	X	X						
T.Misuse					X	X						
T.Crypto								X				
T.Intercept									X			
T.WrongRights			X									
A.Plat-COS										X		
A.Resp-ObjS											X	
P.Process-TOE												X

Table 21 – Security objectives / Threats-Assumptions-Policies correspondence analysis

The following text describes for every OSP, Threat and Assumption, how they are covered by Security Objectives.

The thread **T.Forge_Internal_Data** addresses the falsification of internal user data or TSF data by an attacker. This is prevented by O.Integrity that ensures the integrity of user data, the security services and the TSF data. Also, O.Resp-COS addresses this thread because the user data and TSF data are treated by the TOE as defined by the TSF data of the object system.

The thread **T.Compromise_Internal_Data** addresses the disclosure of confidential user data or TSF data by an attacker. The objective O.Resp-COS requires that the user data and TSF data are treated by the TOE as defined by the TSF data of the object system. Hence, the confidential data are handled correctly by the TSF. The security objective O.Confidentiality ensures the confidentiality of private keys and other confidential TSF data. O.KeyManagement requires that the used keys to protect the confidentiality are generated, imported, distributed, managed and destroyed in a secure way.

The thread **T.Malicious_Application** addresses the modification of user data or TSF data by the installation and execution of a malicious code by an attacker. The security objective O.TSFDataExport requires the correct export of TSF data in order to prevent the export of code fragments that could be used for analysing and modification of TOE code. O.Authentication enforces user authentication in order to control the access protected functions that could be (mis)used to install and execute malicious code. Also, O.AccessControl requires the TSF to enforce an access control policy for the access to restricted objects in order to prevent unauthorised installation of malicious code.

The thread **T.Misuse** addresses the usage of access control protected assets by an attacker without knowledge of user authentication data or by any implicit authorization. This is prevented by the security objective O.AccessControl that requires the TSF to enforce an access control policy for the access to restricted objects. Also the security objective O.Authentication requires user authentication for the use of protected functions.

The thread **T.Crypto** addresses a cryptographic attack to the implementation of cryptographic algorithms or the guessing of keys using brute force attacks. This thread is directly covered by the security objective O.Crypto which requires a secure implementation of cryptographic algorithms.

The thread **T.Intercept** addresses the interception of the communication between the TOE and an external entity by an attacker. The attacker tries to delete, add or forge transmitted data. This thread is directly addressed by the security objective O.SecureMessaging which requires the TOE to establish a trusted channel that protects the confidentiality and integrity of the transmitted data between the TOE and an external entity.

The thread **T.WrongRights** addresses the compromising or manipulation of sensitive user data or TSF data by using undocumented or inappropriate access rights defined in the object system. This thread is addressed by the security objective O.Resp-COS which requires the TOE to treat the user data and TSF data as defined by the TSF data of the object system. Hence the correct access rights are always used and prevent misuse by undocumented or inappropriate access rights to that data.

The assumption **A.Plat-COS** assumes that the object system of the TOE is designed according to dedicated guidance documents and according to relevant findings of the TOE evaluation reports. This assumption is directly addressed by the security objective for the operational environment OE.Plat-COS.

The assumption **A.Resp-ObjS** assumes that all user data and TSF data are treated by the object system as defined for its specific application context. This assumption is directly addressed by the security objective for the operational environment OE.Resp-ObjS.

The OSP **P.Process-TOE** addresses the protection during TOE development and production as defined in BSI-CC-PP-0035-2007 .This is supported by the security objective for the operational environment OE.Process-Card that addresses the TOE after the delivery for phase 5 up to 7: It requires that end consumers maintain the confidentiality and integrity of the TOE and its manufacturing and test data.

6. EXTENDED COMPONENT DEFINITION

6.1 FCS_RNG GENERATION OF RANDOM NUMBERS

To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

The family “Generation of random numbers (FCS_RNG)” is specified as follows.

FCS_RNG Generation of random numbers

Family behaviour

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

Component levelling:

FCS_RNG.1 Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

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.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic,*
hybrid physical, hybrid deterministic] 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*].

6.2 FIA_API AUTHENTICATION PROOF OF IDENTITY

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

Application note: The other families of the Class FIA describe only the authentication verification of users' identity performed by the TOE and do not describe the functionality of the user to prove their identity. The following paragraph defines the extended family FIA_API from point of view of a TOE proving its identity.

FIA_API Authentication Proof of Identity

Family Behaviour

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

Component levelling:

FIA_API.1 Authentication Proof of Identity, provides prove of the identity of the TOE to an external entity.

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

Audit: There are no actions defined to be auditable

FIA_API.1 Authentication Proof of Identity
Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide a [assignment: *authentication mechanism*] to prove the identity of the [assignment: object,*authorized user or role*] to an external entity.

6.3 FPT_EMS TOE Emanation

The 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 secret data stored in and used by the TOE 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, etc. This family describes the functional requirements for the limitation of intelligible emanations being not directly addressed by any other component of CC part 2

FPT_EMS TOE Emanation

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling:

FPT_EMS TOE Emanation
1

FPT_EMS.1 Emanation of TSF and User data, defines limits of TOE emanation related to TSF and User data.

Management: There are no management activities foreseen.

Audit: There are no actions defined to be auditable.
FPT_EMS.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.

FPT_EMS.1.2 Interface Emanation requires not emit interface emanation enabling access to TSF data or user data.

FPT_EMS.1 Emanation of TSF and User data

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*].

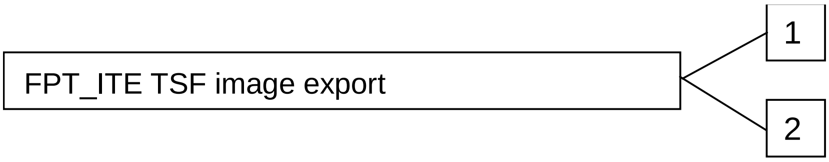
FPT_EMS.1.2 The TSF shall ensure [assignment: *type of users*] are unable to use the following interface [assignment: *type of connection*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

6.4 FPT_ITE TSF IMAGE EXPORT

Family behaviour

The family FPT_ITE (TSF image export) of the class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. This family defines rules for export of TOE implementation fingerprints and of TSF data in order to allow verification of the correct implementation of the IC Dedicated Software and the COS of the TOE and the TSF data of the smartcard. The export of a fingerprint of the TOE implementation, e.g. a keyed hash value over all implemented executable code, provides the ability to compare the implemented executable code with the known intended executable code. The export of all non-confidential TSF data, e.g. data security attributes of subjects and objects and public authentication verification data like public keys, provides the ability to verify their correctness e.g. against a object system specification. The exported data must be correct, but do not need protection of confidentiality or integrity if the export is performed in a protected environment. This family describes the functional requirements for unprotected export of TSF data and export of TOE implementation fingerprints not being addressed by any other component of CC part 2.

Component levelling:



FPT_ITE.1 Export of TOE implementation fingerprint, provides the ability to export the TOE implementation fingerprint without protection of confidentiality or integrity.

FPT_ITE.2 Export of TSF data, provides the ability to export the TSF data without protection of confidentiality or integrity.

Management: FPT_ITE.1, FPT_ITE.2

There are no management activities foreseen.

Audit: FPT_ITE.1, FPT_ITE.2

There are no actions defined to be auditable.

FPT_ITE.1 Export of TOE implementation fingerprint

Hierarchical to: No other components.

Dependencies : No dependencies

FPT_ITE.1.1 The TOE shall export fingerprint of TOE implementation given the following conditions [assignment: *conditions for export*].

FPT_ITE.1.2 The TSF shall use [assignment: *list of generation rules to be applied by TSF*] for the exported data.

FPT_ITE.2 Export of TSF data

Hierarchical to: No other components.

Dependencies : No dependencies

FPT_ITE.2.1 The TOE shall export [assignment: *list of types of TSF data*] given the following

conditions [assignment: *conditions for export*].

FPT_ITE.2.2 The TSF shall use [assignment: *list of encoding rules to be applied by TSF*] for the exported data.

7. TOE SECURITY REQUIREMENTS RATIONALE

7.1 GENERAL

This chapter gives the security functional requirements and the security assurance requirements for the TOE. TOE Security Functional Requirements

The refinement operation is used to add detail to a requirement, and, thus, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in **bold text** and removed words are ~~crossed out~~. In some cases a interpretation refinement is given. In such a case a extra paragraph starting with "Refinement" is given.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted as ***bold underlined and italicized text***.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments are denoted by showing as underlined text.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

7.2 SECURITY FUNCTIONAL REQUIREMENTS FOR THE TOE

7.2.1 Overview

Security Functional Groups	Security Functional
Protection against Malfunction	FRU_FLT.2/SICP, FPT_FLS.1/SICP
Protection against Abuse of Functionality	FMT_LIM.1/SICP, FMT_LIM.2/SICP, FAU_SAS.1/SICP
Protection against Physical Manipulation and Probing	FPT_PHP.3/SICP
Protection against Leakage	FDP_ITT.1/SICP, FPT_ITT.1/SICP, FDP_IFC.1/SICP
Generation of Random Numbers	FCS_RNG.1/SICP

Table 22 – Security functional groups vs. SFRs related to the IC platform

Security Functional Groups	Security Functional Requirements concerned
General Protection of User data and TSF data	FDP_RIP.1, FDP_SDI.2/ReadEF, FDP_SDI.2/Internal, FPT_FLS.1, FPT_EMS.1, FPT_TDC.1, FPT_ITE.1, FPT_ITE.2, FPT_TST.1
Authentication	FIA_AFL.1/PIN, FIA_AFL.1/PUC, FIA_ATD.1, FIA_SOS.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_UID.1, FIA_API.1, FMT_SMR.1, FIA_USB.1
Access Control	FDP_ACC.1/EF, FDP_ACF.1/EF, FDP_ACC.1/MF_DF, FDP_ACC.1/TEF, FDP_ACF.1/TEF, FDP_ACC.1/SEF, FDP_ACF.1/SEF, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FDP_ACC.1/PKEYS, FDP_ACF.1/MF_DF, FDP_ACF.1/PKEYS, FDP_ITC.1/PKEYS, FMT_MSA.3, FMT_SMF.1, FMT_MSA.1/Life, FMT_MSA.1/SEF, FMT_MTD.1/PIN, FMT_MSA.1/PIN, FMT_MTD.1/Auth, FMT_MSA.1/Auth, FMT_MTD.1/NE, FMT_MTD.1/Init, FMT_MTD.1/Perso
Cryptographic Functions	FCS_RNG.1, FCS_COP.1/SHA, FCS_COP.1/COS.3TDES, FCS_COP.1/COS.RMAC, FCS_CKM.1/3TDES_SM, FCS_COP.1/COS.AES, FCS_CKM.1/AES.SM, FCS_CKM.1/RSA, FCS_CKM.1/ELC, , FCS_COP.1/COS.CMAC, FCS_COP.1/COS.RSA.S, FCS_COP.1/COS.RSA.V, FCS_COP.1/COS.ECDSA.S, FCS_COP.1/COS.ECDSA.V, FCS_COP.1/COS.RSA, FCS_COP.1/COS.ELC, FCS_COP.1/PAUTH , FCS_CKM.4
Protection of communication	FTP_ITC.1/TC

Table 23 – Security functional groups vs. SFRs

The original SFRs from the PP in blue font do not appear in the corresponding table 12 of the PP, but are amended here for completeness.

The following TSF Data are defined for the IC part of the TOE.

TSF Data	Definition
TOE pre-personalisation data	Any data supplied by the Card Manufacturer that is injected into the non-volatile memory by the Integrated Circuits manufacturer.
TOE initialisation data	Initialisation Data defined by the TOE Manufacturer to identify TOE and to keep track of the Security IC's production and further life-cycle phases are considered as belonging to the TSF data.

Table 24 – TSF Data defined for the IC part

7.2.2 Users, subjects and objects

The security attributes of human users are stored in password objects. The human user selects the password object by *pwIdentifier* and therefore the role gained by the subject acting for this human user after successful authentication. The role is a set of access rights defined by the access control rules of the objects containing this *pwIdentifier*. The *secret* is used to verify the authentication attempt of the human user providing the authentication verification data. The security attributes *transportStatus*, *lifeCycleStatus* and *flagEnabled* stored in the password object define the status of the role associated with the password. E.g. if the *transportStatus* is equal to *Leer-PIN* or *Transport-PIN* the user is enforced to define his or her own password and making this password and this role effective (by changing the *transportStatus* to *regularPassword*). The multi-reference password shares the *secret* with the password identified by *pwReference.* It allows enforcing re-authentication for access and limitation of authentication status to specific objects and makes password management easier by using the same secret for different roles. The security attributes *interfaceDependentAccessRules*, *startRetryCounter*, *retryCounter*, *minimumLength* and *maximumLength* are defined for the *secret*. The PUC defined for the *secret* is intended for password management and the authorization gained by successful authentication is limited to the command RESET RETRY COUNTER for reset of the *retryCounter* and setting a new *secret.*

The following table provides an overview of the authentication reference data and security attributes of human users and the security attributes of the authentication reference data as TSF data.

User type	Authentication reference data and security attributes	Comments
Human user	**Password** Authentication reference data *Secret* Security attributes of the user role *pwIdentifier* *transportStatus* *lifeCycleStatus* *flagEnabled* *startSsecList* Security attributes of the secret *interfaceDependentAccessRules* *startRetryCounterf* *retryCounter* *minimumLength* *maximumLength*	The following command is used by the TOE to authenticate the human user and to reset the security attribute *retryCounter* by PIN: VERIFY, The following command is used by the TOE to manage the authentication reference data *secret* and the security attribute *retryCounter* with authentication of the human user by PIN: CHANGE REFERENCE DATA (P1='00'). , The following commands are used by the TOE to manage the authentication reference data *secret* without authentication of the human user CHANGE REFERENCE DATA (P1='01') and RESET RETRY COUNTER (P1='02'). The following command is used by the TOE to manage the security attribute *retryCounter* of the authentication reference data PIN without authentication of the human user: RESET RETRY COUNTER (P1='03'). The command GET PIN STATUS is used to query the security attribute *retryCounter* of the authentication reference data PIN with password object specific access control rules. The following commands are used by the TOE to manage the security attribute *flagEnabled* of the authentication reference data with human user authentication by PIN: ENABLE VERIFICATION REQUIREMENT (P1='00') , DISABLE VERIFICATION REQUIREMENT (P1='00'). The following commands are used by the TOE to manage the security attribute *flagEnabled* of the authentication reference data without human user authentication: ENABLE VERIFICATION REQUIREMENT (P1='01'), DISABLE VERIFICATION REQUIREMENT (P1='01'). The commands , ACTIVATE, DEACTIVATE, , and TERMINATE are used to manage the security attribute *lifeCycleStatus* of the authentication reference data password with password object specific access control rules. The command DELETE is used to delete the authentication reference data password with password object specific access control rules.

User type	Authentication reference data and security attributes	Comments
Human user	**Multi-Reference password** Authentication reference data *Secret* is shared with the password identified by *pwReference.* Security attributes of the user role *pwIdentifier,* *lifeCycleStatus,* *transportStatus* *flagEnabled* *startSsecList.* Security attributes of the secret The security attributes *interfaceDependentAccessRules, minimumLength, maximumLength, startRetryCounter* and *retryCounter* are shared with password identified by *pwReference.*	The commands used by the TOE to authenticate the human user and to manage the authentication reference Multi-Reference password data are the same as for password.
Human user	**Personal unblock code (PUC)** Authentication reference data *PUK* Security attributes *pwIdentifier* of the password[2], *pukUsage*	The following command is used by the TOE to manage the authentication reference data *secret* and the security attribute *retryCounter* of the authentication reference data PIN with authentication of the human user by PUC: RESET RETRY COUNTER (P1='00'). The following command is used by the TOE to manage the security attribute *retryCounter* of the authentication reference data PIN with authentication of the human user by PUC: RESET RETRY COUNTER (P1='01').

Table 25 – Authentication reference data of the human user and security attributes

[2] The PUC is part of the password object as authentication reference data for the RESET RETRY COUNTER command for this password.

The security attributes of devices depend on the authentication mechanism and the authentication reference data. A device may be associated with a symmetric cryptographic authentication key with a specific *keyIdentifier* and therefore the role gained by the subject acting for this device after successful authentication. The role is defined by the access control rules of the objects containing this *keyIdentifier*. A device may be also associated with a certificate containing the public key as authentication reference data and the card holder authorization (*CHA*)
in case of RSA-based CVC or the card holder authorization template (*CHAT*) in case of ELC based CVC.
. The authentication protocol comprise the verification of the certificate by means of the root public key and command PSO VERIFY CERTIFICATE and by means of the public key contained in the successful verified certificate and the command EXTERNAL AUTHENTICATE. The subject acting for this device get the role of the *CHA or CHAT* which is referenced in the access control rules of the objects. The security attribute *lifeCycleStatus* is defined for persistently stored keys only.

User type	Authentication reference data and security attributes	Comments
Device	**Symmetric authentication key** Authentication reference data *macKey*[3] Security attributes of the Authentication reference data *keyIdentifier* *interfaceDependentAccessRules* *lifeCycleStatus* *algorithmIdentifier* *numberScenario*	The following commands are used by the TOE to authenticate a device EXTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE, and GENERAL AUTHENTICATE The following commands are used by the TOE to manage the authentication reference data ACTIVATE, DEACTIVATE, DELETE and TERMINATE
Device	**Asymmetric authentication key** Authentication reference data *Root Public Key* *Certificate* containing the *public key* of the device[4] *persistentCache* , *applicationPublicKeyList*[5] Security attributes of the user *Certificate Holder Reference* (*CHR*) *lifeCycleStatus*, *interfaceDependentAccessRules*, *Certificate Holder Authorization* (*CHA*) for RSA keys or *Certificate Holder Authorization Template (CHAT)* for elliptic curve keys Security attributes in the *certificate* *Certificate Profile Identifier (CPI)* *Certification Authority Reference (CAR)* *Object Identifier (OID)*	The following command is used by the TOE to authenticate a device EXTERNAL EXTERNAL AUTHENTICATE with *algID* equal to *rsaRoleCheck* or *elcRoleCheck* The following commands are used by the TOE to manage the authentication reference data PSO VERIFY CERTIFICATE, ACTIVATE, DEACTIVATE, DELETE and TERMINATE

[3] The symmetric authentication object contains encryption key *encKey* and a message authentication key *macKey*.

[4] The certificate of the device may be only end of a certificate chain going up to the root public key.

[5] The command PSO VERIFY CERTIFICATE may store the successful verified public key temporarily in the *volatileCache*or persistenly in the *applicationPublicKeyList*. Public keys in the *applicationPublicKeyList* may be used like root public keys. The wrapper specification and COS specification define the attribute *persistentPublicKeyList* as superset of all persistently stored public key in the *applicationPublicKeyList* and the *persitentCache*.

User type	Authentication reference data and security attributes	Comments
Device	**Secure messaging channel key** Authentication reference data MAC session key SK4SM Security attributes of SK4SM *flagSessionEnabled* equal SK4SM, *Kmac* and *SSCmac*, *negotiationKeyInformation*.	The TOE authenticates the sender of a received command using secure messaging

Table 26 – Authentication reference data of the devices and security attributes

The following table defines the authentication verification data used by the TSF itself for authentication by external entities (cf. FIA_API.1).

Subject type	Authentication verification data and security attributes	Operations
TSF	**Private authentication key** Authentication verification data *privateKey* Security attributes *keyIdentifier* *setAlgorithmIdentifier* with *algorithmIdentifier* *lifeCycleStatus*	The following commands are used by the TOE to authenticate themselves to an external device: INTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE
TSF	**Secure messaging channel key** Authentication verification data MAC session key SK4SM Security attributes *flagSessionEnabled*, *macKey* and *SSCmac*, *encKey* and *SSCenc*, *flagCmdEnc* and *flagRspEnc*	Responses using secure messaging. The session keys are linked to the folder of the keys used to them.

Table 27 – Authentication verification data of the TSF and security attributes

In usage phase the COS specification associates a subject with a *logical channel* and its *channelContext* The TOE supports one subject respective logical channel, Package Logical Channel is not implemented. The *channelContext* comprises security attributes of the subject summarized in the following table.

Security attribute	Elements	Comments
interface		The TOE detects whether the communication uses contact based interface (value set to *kontaktbehaftet*).The TOE behaves as *interfaceDependentAccess* *Rules* is permanently set to “*kontaktbehaftet”*.
currentFolder		Identifier of the (unique) current folder
	seIdentifier	Security environment selected by means of command MANAGE SECURITY ENVIRONMENT. If no security environment is explicitly selected the default security environment #1 is assumed.
keyReferenceList		The list contains elements which may be empty or may contain one pair (*keyReference*, *algorithmIdentifier).*
	externalAuthenticate	*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for device authentication by means of commands EXTERNAL AUTHENTICATE and MUTUAL AUTHENTICATE
	internalAuthenticate	*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for authentication of the TSF itself by means of commands INTERNAL AUTHENTICATE
	verifyCertificate	*keyReference* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for PSO VERIFY CERTIFICATE
	signatureCreation	*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for PSO COMPUTE DIGITAL SIGNATURE
	dataDecipher	*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for PSO DECIPHER or PSO TRANSCIPHER
	dataEncipher	*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for PSO ENCIPHER.

Security attribute	Elements	Comments
	~~*macCalculation*~~	~~*keyReference* and *algorithmIdentifier* of the key selected by means of the command MANAGE SECURITY ENVIRONMENT to be used for PSO COMPUTE CRYPTOGRAPHIC CHECKSUM and PSO VERIFY CRYPTOGRAPHIC CHECKSUM if package Crypto Box is supported.~~ [6]
SessionkeyContext		This list contains security attributes associated with secure messaging and trusted channels.
	flagSessionEnabled	Value *noSK* indicates no session key established. Value *SK4SM* indicates session keys established for receiving commands and sending responses. ~~Value *SK4TC* indicates session keys established for PSO COMPUTE CRYPTOGRAPHIC CHECKSUM, PSO VERIFY CRYPTOGRAPHIC CHECKSUM and PSO ENCIPHER, PSO DECIPHER if package Crypto Box is supported.~~[7]
	encKey and *SSCenc*	Key for encryption and decryption and its sequence counter
	macKey and *SSCmac*	Key for MAC calculation and verification and its sequence counter
	flagCmdEnc and *flagRspEnc*	Flags indicating encryption of data in commands respective responses
	negotiationKeyInformation	*keyIdentifier* of the key used to generate the session keys and if asymmetric key was used the *accessRigth* associated with this key. ~~The keyIdentifier may reference to the authentication reference data used for PACE if PACE is supported by the TOE.~~ [8]
	accessRulesSession-keys	Access control rules associated with trusted channel support .
globalPasswordList	(*pwReference*, *securityStatusEvaluationCounter*)	List of 0, 1, 2, 3 or 4 elements containing results of successful human user authentication with password in MF: *pwReference* and *securityStatusEvaluationCounter*
dfSpecificPasswordList	(*pwReference*, *securityStatusEvaluationCounter*)	List of 0, 1, 2, 3 or 4 elements containing results of successful human user authentication with password for each DF: *pwReference* and *securityStatusEvaluationCounter*

[6] PSO COMPUTE/VERIFY CRYPTOGRAPHIC CHECKSUM suppressed vompared to the PP, because not supported.

[7] KS4SM suppressed compared to the PP, because not supported.

[8] PACE suppressed compared to the PP, because not supported.

Security attribute	Elements	Comments
globalSecurityList	*CHA* or *keyIdentifier*	List of 0, 1, 2 or 3 elements containing results of successful device authentication with authentication reference data in MF: *CHA* as reference to the role gained by authentication based on certificate or *keyIdentifier* as reference to the used symmetric authentication key ~~or *keyIdentifier* generated by successful authentication with PACE protocol if PACE is supported by the TOE .~~[9]
dfSpecificSecurityList	*CHA* or *keyIdentifier*	List of 0, 1, 2 or 3 elements containing results of successful device authentication with authentication reference data for each DF: *CHA* *CHA* as reference to the role gained by authentication based on certificate or *keyIdentifier* as reference to symmetric authentication key ~~or *keyIdentifier* generated by successful authentication with PACE protocol if PACE is supported by the TOE .~~[10]
bitSecurityList		List of CHAT gained by successful authentication with CVC based on ECC. The effective access rights are the intersection of access rights defined in CVC of the CVC chain up to the root.
Current file		Identifier of the (unique) current file from *currentFolder.children*
securityStatusEvaluationCounter	*startSsec*	Must contain all values of *startSsec* and may be *empty*

Table 28 – Security attributes of a subject

The following tables provide an overview of the objects, operations and security attributes. All references in the table refer to the technical specification of the card operating system. The security attribute *lifeCycleStatus* is defined for persistently stored keys only.

[9] PACE suppressed compared to the PP, because not supported.

[10] PACE suppressed compared to the PP, because not supported.

Object type	Security attributes	Operations
Object system	*applicationPublicKeyList,* *persistentCache, pointInTime*	PSO VERIFY CERTIFICATE
Folder	*accessRules:* *lifeCycleStatus* ~~*shareable*[11]~~ *interfaceDependentAccessRules* *children*	SELECT ACTIVATE DEACTIVATE DELETE FINGERPRINT ~~GET RANDOM[12]~~ LOAD APPLICATION TERMINATE DF
Dedicated File	Additionally to Folder: *fileIdentifier*	Identical to Folder
Application	Additionally to Folder: *applicationIdentifier*	Identical to Folder
Application Dedicated File	Additionally to Folder: *fileIdentifier* *applicationIdentifier children*	Identical to Folder
Elementary File	*fileIdentifier* *list of shortFileIdentifier* *lifeCycleStatus* ~~*shareable*[13]~~ *accessRules:* *interfaceDependentAccessRules* *flagTransactionMode* *flagChecksum*	SELECT ACTIVATE DEACTIVATE DELETE TERMINATE
Transparent EF	Additionally to Elementary File: *numberOfOctet* *positionLogicalEndOfFile* *body*	Additionally to Elementary File: ERASE BINARY READ BINARY UPDATE BINARY WRITE BINARY

[11] ~~Available with package logical channel~~
[12] ~~Only available with package crypto box~~
[13] ~~Available with package logical channel~~

Object type	Security attributes	Operations
Structured EF	Additionally to Elementary File: *recordList* *maximumNumberOfRecords* *maximumRecordLength* *flagRecordLifeCycleStatus*	Additionally to Elementary File: ACTIVATE RECORD APPEND RECORD DELETE RECORD DEACTIVATE RECORD ERASE RECORD READ RECORD SEARCH RECORD SET LOGICAL EOF UPDATE RECORD
Regular Password (PIN)	*lifeCycleStatus* *pwdIdentifier* *accessRules:* *interfaceDependentAccessRules* *secret: PIN* *minimumLength* *maximumLength* *startRetryCounter* *retryCounter* *transportStatus* *flagEnabled* *startSsecList* *PUC* *pukUsage* channel specific: *securityStatusEvaluationCounter*	ACTIVATE DEACTIVATE DELETE TERMINATE CHANGE REFERENCE DATA DISABLE VERIFICATION REQUIREMENT ENABLE VERIFICATION REQUIREMENT GET PIN STATUS RESET RETRY COUNTER VERIFY

Object type	Security attributes	Operations
Multi-reference Password (MR-PIN)	*lifeCycleStatus* *pwdIdentifier* *accessRules:* *interfaceDependentAccessRules* *startSsecList* *flagEnabled* *passwordReference* Attributes used together with *referred password (PIN):* *secret: PIN* *minimumLength* *maximumLength* *startRetryCounter* *retryCounter* *transportStatus* *PUC* *pukUsage* channel specific: *securityStatusEvaluationCounter*	Identical to Regular Password
PUC	*type pin* *pukUsage*	RESET RETRY COUNTER
Symmetric Key	*lifeCycleStatus* *keyIdentifier* *accessRules:* *interfaceDependentAccessRules* *encKey* *macKey* *numberScenario* *algorithmIdentifier* *accessRulesSessionkeys:* *interfaceDependentAccessRules*	ACTIVATE DEACTIVATE DELETE TERMINATE EXTERNAL AUTHENTICATE GENERAL AUTHENTICATE INTERNAL AUTHENTICATE MUTUAL AUTHENTICATE

Object type	Security attributes	Operations
Private Asymmetric Key	*lifeCycleStatus* *keyIdentifier* accessRules: *interfaceDependentAccessRules* *privateKey* *listAlgorithmIdentifier* *accessRulesSessionkeys:* *interfaceDependentAccessRules* *algorithmIdentifier* *keyAvailable*	ACTIVATE DEACTIVATE DELETE TERMINATE GENERATE ASYMMETRIC KEY PAIR or key import EXTERNAL AUTHENTICATE GENERAL AUTHENTICATE INTERNAL AUTHENTICATE PSO COMPUTE DIGITAL SIGNATURE PSO DECIPHER PSO TRANSCIPHER
Public Asymmetric Key	*lifeCycleStatus* *keyIdentifier* *oid* accessRules: *interfaceDependentAccessRules*	ACTIVATE DEACTIVATE DELETE TERMINATE
Public Asymmetric Key for signature verification	Additionally to Public Asymmetric Key: *publicRsaKey: oid* or *publicElcKey: oid* *CHAT* *expirationDate: date*	Additionally to Public Asymmetric Key: PSO VERIFY CERTIFICATE, PSO VERIFY DIGITAL SIGNATURE
Public Asymmetric Key for Authentication	*publicRsaKey: oid* or *publicElcKey: oid* *CHA* *CHAT* *expirationDate: date*	Additionally to Public Asymmetric Key: EXTERNAL AUTHENTICATE GENERAL AUTHENTICATE INTERNAL AUTHENTICATE
Public Asymmetric Key for Encryption	Additionally to Public Asymmetric Key: *publicRsaKey: oid* *publicElcKey: oid*	Additionally to Public Asymmetric Key: PSO ENCIPHER

Object type	Security attributes	Operations
Rule Object containing a TOE access control rule	*ruleIdentifier* *accessRules: interfaceDependentAccessRules*	LOAD APPLICATION DELETE
Card verifiable certificate (CVC)	Certificate Profile Identifier (CPI) Certification Authority Reference (CAR) Certificate Holder Reference (CHR) Certificate Holder Autorisation (CHA) Object Identifier (OID) signature	

Table 29 –Subjects, objects, operations and security attributes.

Logical channel and crypto box packages are not implemented.

Rule Objects are an extension to [gemSpec_COS] to implement *interfaceDependentAccessRules* not as an intrinsic part of each object, but as separate containers that can be referenced by any object. While there is no functional difference for their original purpose, additional operations are possible: creation, deletion, and modification of Rule Objects. These operations are bound to the commands LOAD APPLICATION and DELETE, evaluating the *interfaceDependentAccessRules* linked to a Rule Object itself, so no security objective gets undermined.

The TOE must support Access control lists for

- *lifeCycleStatus* values *"Operation state(activated)", "Operation state(deactivated)"* and *"Termination state"*,
- *security environments* with value *seIdentifier* selected for the folder, and
- *interfaceDependentAccessRules* for contact based communication .

If the user communicates with the TOE through the contact based interface the security attribute "*interface*" of the subject is set to the value "*kontaktbehaftet"* and the *interfaceDependentAccessRules* for contact based communication shall apply. The TOE does not support the contactless communication; thus it behaves in respect to access control like a TOE defining all *interfaceDependentAccessRules* "*kontaktlos"* set to *NEVER* in the object system.

The user may set the *seIdentifier* value of the *security environments* for the folder by means of the command MANAGE SECURITY ENVIRONMENT. This may be seen as selection of a specific set of access control rules for the folder and the objects in this folder.[14]

The TOE access control rule contains

- command defined by CLA, 0 or 1 parameter P1, and 0 or 1 parameter P2,
- values of the *lifeCycleStatus* and *interfaceDependentAccessRules* indicating the set of access control rules to be applied,
- access control condition defined as Boolean expression with Boolean operators AND and OR of Boolean elements of the following types *ALWAYS*, *NEVER*, *PWD*(*pwIdentifier*), *AUT*(*keyReference*), *AUT*(*CHA*), *AUT*(*CHAT*) and secure messaging.

Note AUT(CHAT) is true if the access right bit necessary for the object and the command is 1 in the effective access rights calculated as bitwise-AND of all CHAT in the CVC chain verified successfully by PSO VERIFY DIGITAL SIGNATURE command executions.

The Boolean element ALWAYS provides the Boolean value TRUE. The Boolean element NEVER provides the Boolean value FALSE. The other Boolean elements provide the Boolean value TRUE if the value in the access control list match its corresponding security attribute of the subject and provides the Boolean value FALSE is they do not match.

[14] This approach is used e.g. for signature creation with eHPC: the signatory selects security environment #1 for single signature, and security environment #2 for batch signature creation requirering additional authentication of the signature creation application

7.2.3 Security Functional Requirements for the TOE taken over from BSI-CC-PP-0035-2007

All SFRs from section"Security Functional Requirements for the TOE" of the BSI-CC-PP-0035-2007 are part of this ST. On all SFR of the BSI-CC-PP-0035-2007 an iteration operation is performed. For the iteration operation the suffix "/SICP" is added to the SFR name from BSI-CC-PP-0035-2007.

The complete list of the SFRs taken over from BSI-CC-PP-0035-2007 follows. For further descriptions, details, and interpretations refer section 6.1 in BSI-CC-PP-0035-2007.

- FRU_FLT.2/SICP: Limited fault tolerance.
- FPT_FLS.1/SICP: Failure with preservation of secure state.
- FMT_LIM.1/SICP: Limited capabilities.
- FMT_LIM.2/SICP: Limited capabilities
- FAU_SAS.1/SICP: Audit storage
- FPT_PHP.3/SICP: Resistance to physical attack.
- FDP_ITT.1/SICP: Basic internal transfer protection.
- FPT_ITT.1/SICP: Basic internal TSF data transfer protection.
- FDP_IFC.1/SICP: Subset information flow control.
- FCS_RNG.1/SICP: Random number generation

Table below maps the SFR name in this ST to the SFR name in BSI-CC-PP-0035-2007 . This approach allows an easy and unambiguous identification which SFR was taken over from the BSI-CC-PP-0035-2007 into this ST and which SFR is defined newly in this ST.

SFR name in this ST	SFR name in BSI-CC-PP-0035-2007
FRU_FLT.2/SICP	FRU_FLT.2
FPT_FLS.1/SICP	FPT_FLS.1
FMT_LIM.1/SICP	FMT_LIM.1
FMT_LIM.2/SICP	FMT_LIM.2
FAU_SAS.1/SICP	FAU_SAS.1
FPT_PHP.3/SICP	FPT_PHP.3
FDP_ITT.1/SICP	FDP_ITT.1
FPT_ITT.1/SICP	FPT_ITT.1
FDP_IFC.1/SICP	FDP_IFC.1
FCS_RNG.1/SICP	FCS_RNG.1

Table 30 –Mapping between SFR names in this ST and the SFR names in the BSI-CC-PP-0035-2007

The TOE shall meet the requirement “Limited fault tolerance (FRU_FLT.2/SICP)” as specified below.

FRU_FLT.2/SICP Limited fault tolerance
Hierarchical to: FRU_FLT.1 Degraded fault tolerance
Dependencies: FPT_FLS.1 Failure with preservation of secure state.
FRU_FLT.2.1/SICP The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/SICP)

The TOE shall meet the requirement “Failure with preservation of secure state (FPT_FLS.1/SICP)” but the FPT_FLS.1 (Composite) includes an equivalent requirement as **FPT_FLS.1/SICP**. So it’s not needed to specified it here (see 7.2.4).

The TOE shall meet the requirement “Limited capabilities (FMT_LIM.1/SICP)” as specified below (Common Criteria Part 2 extended).
FMT_LIM.1/SICP Limited capabilities
Hierarchical to: No other components.
Dependencies: FMT_LIM.2 Limited availability

FMT_LIM.1.1/SICP The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks

.

The TOE shall meet the requirement “Limited availability (FMT_LIM.2/SICP)” as specified below (Common Criteria Part 2 extended).
FMT_LIM.2/SICP Limited availability
Hierarchical to: No other components.
Dependencies: FMT_LIM.1 Limited capabilities.
FMT_LIM.2.1/SICP The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with “Limited capabilities (FMT_LIM.1/SICP)” the following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks

The TOE shall meet the requirement “Audit storage (FAU_SAS.1/SICP)” as specified below (Common Criteria Part 2 extended).

FAU_SAS.1/SICP Audit storage
Hierarchical to: No other components.
Dependencies: No dependencies.
FAU_SAS.1.1/SICP The TSF shall provide the test process before TOE Delivery with the capability to store the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software in the not changeable configuration page area and non-volatile memory

The TOE shall meet the requirement “Resistance to physical attack (FPT_PHP.3)” as specified below.
FPT_PHP.3/SICP Resistance to physical attack
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_PHP.3.1/SICP The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced

The TOE shall meet the requirement “Basic internal transfer protection (FDP_ITT.1/SICP)” as specified below.
FDP_ITT.1/SICP Basic internal transfer protection
Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset accesscontrol, or FDP_IFC.1 Subset information flow control]
FDP_ITT.1.1/SICP The TSF shall enforce the Data Processing Policy to prevent ***the disclosure*** of user data when it is transmitted between physically-separated parts of the TOE.

The TOE shall meet the requirement “Basic internal TSF data transfer protection (FPT_ITT.1/SICP)” as specified below.

FPT_ITT.1/SICP Basic internal TSF data transfer protection
Hierarchical to: No other components.
Dependencies: No dependencie
FPT_ITT.1.1/SICP The TSF shall protect TSF data from ***disclosure*** when it is transmitted between separate parts of the TOE.

The TOE shall meet the requirement " Subset information flow control (FDP_IFC.1/SICP)" as specified below:

FDP_IFC.1/SICP	Subset information flow control
Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/SICP	The TSF shall enforce the Data Processing Policy on all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software

The TOE shall meet the requirement"Quality metric for random numbers (FCS_RNG.1/SICP)" but FCS_RNG.1(composite) includes an equivalent requirement as **FCS_RNG.1/SICP** (PTG2). So it's not needed to specified it here (see 7.2.7).

7.2.4 General Protection of User data and TSF data

The TOE shall meet the requirement "Subset residual information protection (FDP_RIP.1)" as specified below.

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 ***deallocation of the resource from*** the following objects: password objects, secret cryptographic keys, private cryptographic keys, session keys, none

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2/Internal)" as specified below.

FDP_SDI.2/Internal Stored data integrity monitoring and action
Hierarchical to: FDP_SDI.1 Stored data integrity monitoring
Dependencies: No dependencies.

FDP_SDI.2.1/Internal The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all objects, based on the following attributes:
(1) Key objects
(2) PIN objects
(3) *affectedObject.flagTransactionMode=TRUE,*
(4) none

FDP_SDI.2.2/Internal Upon detection of a data integrity error, the TSF shall :
(1) prohibit the use of the altered data and
(2) halt TOE execution.

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2/ReadEF)" as specified below.

FDP_SDI.2/ReadEF Stored data integrity monitoring and action
Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1/ReadEF The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all objects, based on the following attributes:

(1) EF objects, if configured to support integrity checking (flagChecksum)

FDP_SDI.2.2/ReadEF Upon detection of a data integrity error, the TSF shall transmit the data together with a warning status.

The preceding iteration of FDP_SDI.2. is not a functional difference compared to the PP. It is only a categorization of the affected objects with respect to the two possible actions to be taken: TOE halt or response with warning. So neither this nor any other security objective gets undermined.

The TOE shall meet the requirement "Failure with preservation of secure state (FPT_FLS.1)" as specified below.

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) exposure to operating conditions where therefore a malfunction could occur (2) failure detected by TSF according to FPT_TST.1.

The TOE shall meet the requirement “FPT_EMS.1 (FPT_EMS.1)” as specified below .

FPT_EMS.1	Emanation of TSF and User data
Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_EMS.1.1 The TOE shall not emit electromagnetic radiation in excess of unintelligible emission enabling access to the following TSF data

(1) Regular password,
(2) Multi-Refernce password,
(3) PUC,
(4) Session keys,
(5) Symmetric authentication keys,
(6) Private authentication keys,
(7) None
and the *following user data*
(8) Private asymmetric keys,
(9) Symmetric keys,
(10) None

FPT_EMS.1.2 The TSF shall ensure any user are unable to use the following interface circuit interfaces to gain access to the following TSF data

(1) Regular password,
(2) Multi-Reference password,
(3) PUC,
(4) Session keys,
(5) Symmetric authentication keys,
(6) Private authentication keys,
(7) None
and the *following user data*
(8) Private asymmetric keys,
(9) Symmetric keys,
(10) None

The TOE shall meet the requirement “Inter-TSF basic TSF data consistency (FPT_TDC.1)” as specified below.

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret Card Verifiable Certificate (CVC) when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use COS specification chapter 7 Certificate” and append “CV-Certificate for ELC-keys” when interpreting the TSF data from another trusted IT product.

The TOE shall meet the requirement “Export of TOE implementation fingerprint (FPT_ITE.1)” as specified below.

FPT_ITE.1 Export of TOE implementation fingerprint

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITE.1.1 The TOE shall export fingerprint of TOE implementation given the following conditions execution of the command FINGERPRINT (COS specification).

FPT_ITE.1.2 The TSF shall use ***SHA-256 based fingerprint of the TOE implementation*** for the exported data.

The TOE shall meet the requirement “Export of TSF data (FPT_ITE.2)” as specified below.

FPT_ITE.2 Export of TSF data

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITE.2.1 The TOE shall export

(1) all public authentication reference data,

(2) all security attributes of the object system and for all objects of the object system for all commands,

(3) none

given the following conditions

(1) no export of secret data,
(2) no export of private keys,
(3) no export of secure messaging keys,
(4) no export of passwords and PUC.

FPT_ITE.2.2 The TSF shall use the encoding rules defined in the Technical Guidance TR-03143 for the exported data.

Application note: The public TSF data addressed as TSF data in bullet (1) in the element FPT_ITE.2.1 covers at least all root public key and other public keys used as authentication reference data persistent stored in the object system (
(cf. *applicationPublicKeyList* and *persistentCache*) and exported by command LIST PUBLIC KEY The bullet (2) in the element FPT_ITE.2.1 covers all security attributes of the object system and of all objects of object types listed in Table 29 and all TOE specific security attributes and parameters (except secrets). The COS specification identifies optional functionality the TOE may support. The TOE (as COS, wrapper and guidance documentation) must support the user to find all objects and to export all security attributes of these objects. Note while MF, DF and EF are hierarchically structured the Application and Application Dedicated File are directly referenced which may require special methods to find all objects in the object system. Note the *listOfApplication* as security attribute of the object system contains at least one *applicationIdentifier* of each Application or Application Dedicated File. The exported data shall be encoded by wrapper to allow interpretation of the TSF data. The encoding rules shall meet the requirements of the Technical Guidance TR-03143 describing the verification tool used for examination of the object system against the specification of the object system.

The TOE shall meet the requirement “TSF testing (FPT_TST.1)” as specified below.

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 **(1) during initial start-up and,** **(2) before critical operations** to demonstrate the correct operation of ***the TSF.***
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of ***TSF data.***
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of ***TSF***.

7.2.5 Authentication

The TOE shall meet the requirement “Verification of secrets (FIA_SOS.1)” as specified below.

FIA_SOS.1	Verification of secrets
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_SOS.1.1	The TSF shall provide a mechanism to verify that secrets **provided by the user for password objects** meet the quality metric: length not lower than *minimumLength* and not greater than *maximumLength*

The TOE shall meet the requirement “Authentication failure handling (FIA_AFL.1/PIN)” as specified below.

FIA_AFL.1/PIN	Authentication failure handling
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication.
FIA_AFL.1.1/PIN	The TSF shall detect when ~~an administrator~~ configurable positive integer within 1 to 15 unsuccessful authentication attempts occur related to consecutive failed human user authentication for the PIN via VERIFY, ENABLE VERIFICATION REQUIREMENT, DISABLE VERIFICATION REQUIREMENT or CHANGE REFERENCE DATA command

FIA_AFL.1.2/PIN When the defined number of unsuccessful authentication attempts has been ***met***, the TSF shall block the password for authentication until successful unblock using command RESET RETRY COUNTER

(1) P1=’00’ or P1=’01’ with presenting unblocking code PUC of this password object,

(2) P1=’02’ or P1=’03’ without presenting unblocking code PUC of this password object .

Application note: The component FIA_AFL.1/PIN addresses the human user authentication by means of a password. The configurable positive integer of unsuccessful authentication attempts is defined in the password objects of the object system.”Consecutive failed authentication attemps” are counted separately for each PIN and interrupted by successful authentication attempt for this PIN, i.e. the PIN object has a retryCounter wich is initially set to startRetryCounter, decremented by each failed authentication attempt and reset to startRetryCounter by successful authentication with the PIN or be successful execution of the command RESET RETRY COUNTER. The command RESET RETRY COUNTER (CLA,INS,P1)=(00,2C,02) and (CLA,INS,P1)=(00,2C,03) unblock the PIN without presenting unblocking code PUC of this password object. In order to prevent bypass of the human user authentication defined by the PIN or PUC the object system shall define access control to this command as required by the security needs of the specific application context, cf. OE.Resp-ObjS.

The TOE shall meet the requirement “Authentication failure handling (FIA_AFL.1/PUC)” as specified below.

FIA_AFL.1/PUC	Authentication failure handling
Hierarchical to:	No other components
Dependencies:	FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1/PUC

The TSF shall detect when ~~an administrator~~ configurable positive integer within 1 to 15 ~~unsuccesful~~ authentication attempts occur related to usage of a password unblocking code using the RESET RETRY COUNTER command

FIA_AFL.1.2/PUC When the defined number of ~~unsuccessful~~ authentication attempts has been ***met***, the TSF shall block the password unblocking code.

The TOE shall meet the requirement “User attribute definition (FIA_ATD.1)” as specified below.

FIA_ATD.1	User attribute definition
Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

(1) for Human User: authentication state gained

a. with password: *pwdIdentifier* in *globalPasswordList* and *pwdIdentifier* in *dfSpecificPasswordList*,

b. with Multi-Reference password: *pwIdentifier* in *globalPasswordList* and *pwIdentifier* in *dfSpecificPasswordList*,

(2) for Device: authentication state gained

a. by CVC with CHA in *globalSecurityList* if CVC is stored in MF and *dfSpecificSecurityList* if CVC is stored in a DF,

b. by CVC with CHAT in bitSecurityList,

c. with symmetric authentication key: keyIdentity of the key,

d. with secure messaging keys: keyIdentity of the key used for establishing the session key.

The TOE shall meet the requirement "Timing of authentication (FIA_UAU.1)" as specified below.

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) reading the ATR,

~~(2) [selection: *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*]~~[15]

(3) commands with access control rule ALWAYS for the current life cycle status and depending on the interface,

(4) none

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.

[15] The TOE defines TOE specific access control rules for all commands, including GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT and SELECT, cf. COS specification. Therefore they all fall under case (3), and case (2) is removed as not applicable

Application note: ATR means Cold ATR and Warm ATR (cf. COS specification (N019.900)b). The TOE defines access control limitation for the commands GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT and SELECT, cf. COS specification (N022.810).

The TOE shall meet the requirement "Single-use authentication mechanisms (FIA_UAU.4)" as specified below.

FIA_UAU.4	Single-use authentication mechanisms
Hierarchical to:	No other components.
Dependencies:	No dependencies.

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

(1) external device authentication by means of executing the command EXTERNAL AUTHENTICATE with symmetric or asymmetric key,

(2) external device authentication by means of executing the command MUTUAL AUTHENTICATE with symmetric or asymmetric key,

(3) external device authentication by means of executing the command GENERAL AUTHENTICATE with symmetric or asymmetric key.

(4) None

The TOE shall meet the requirement “Multiple authentication mechanisms (FIA_UAU.5)” as specified below.

FIA_UAU.5	Multiple authentication mechanisms
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1	The TSF shall provide (1) the execution of the VERIFY command, (2) the execution of the CHANGE REFERENCE DATA command, (3) the execution of the RESET RETRY COUNTER command, (4) the execution of the EXTERNAL AUTHENTICATE command, (5) the execution of the MUTUAL AUTHENTICATE command, (6) the execution of the GENERAL AUTHENTICATE command, (7) a secure messaging channel, (8) a trusted channel to support user authentication
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to the following rules: (1) password based authentication shall be used for authenticating a human user by means of commands VERIFY, CHANGE REFERENCE DATA and RESET RETRY COUNTER, (2) key based authentication mechanisms shall be used for authenticating of devices by means of commands EXTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE and GENERAL AUTHENTICATE, (3) none.

The TOE shall meet the requirement "Re-authenticating (FIA_UAU.6)" as specified below:.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1 The TSF shall re-authenticate the ~~user~~ **sender of a message** under the conditions

(1) each command sent to the TOE after establishing the secure messaging by successful authentication after execution of the INTERNAL AUTHENTICATE and EXTERNAL AUTHENTICATE, or MUTUAL AUTHENTICATE or GENERAL AUTHENTICATE commands shall be verified as being sent by the authenticated device,

The TOE shall meet the requirement "Timing of identification (FIA_UID.1)" as specified below.

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow

(1) reading the ATR

~~(2)~~ [16]~~[selection: *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*]~~

(3) commands with access control rule ALWAYS for the current life cycle status and depending on the interface,

(4) None

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

Application note: The TOE defines access control limitation for the commands GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT and SELECT, cf. COS specification (N022.810).

[16] The TOE defines TOE specific access control rules for all commands, including GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT and SELECT, cf. COS specification. Therefore they all fall under case (3), and case (2) is removed as not applicable

The TOE shall meet the requirement “Authentication Proof of Identity (FIA_API.1)” as specified below (Common Criteria Part 2 extended (see section 5.1)).

FIA_API.1 Authentication Proof of Identity

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide

(1) INTERNAL AUTHENTICATE
(2) MUTUAL AUTHENTICATE
(3) GENERAL AUTHENTICATE,

to prove the identity of the TSF itself to an external entity

The TOE shall meet the requirement “Security roles (FMT_SMR.1)” as specified below:

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

(1) World as unauthenticated user without authentication reference data,
(2) Human User authenticated by password in the role defined for this password,
(3) Human User authenticated by PUC as holder of the corresponding password,
(4) Device authenticated by means of symmetric key in the role defined for this key,
(5) Device authenticated by means of asymmetric key in the role defined by the Certificate Holder Authorisation in the CVC,
(6) Initializer authenticated by K_{ICC},
(7) Personalizer authenticated by Perso Keys,
(8) None .

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

The TOE shall meet the requirement “User-subject binding (FIA_USB.1)” as specified below.

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

(1) For Human User authenticated with password: *pwIdentifier* and Authentication Context *globalPasswordList* and *dfSpecificPasswordList*.

(2) For Human User authenticated with PUC: *pwIdentifier* of corresponding password,

(3) For Device the Role authenticated by RSA based CVC : the Certificate Holder Authorisation (CHA) in the CVC

(4) For Device the Role authenticated by ECC based CVC: the Certificate Holder Authorisation Template (CHAT),

(5) For Device the Role authenticated by symmetric key: *keyIdentifier* and Authentication Context.

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

(1) If the logical channel is reset by command Manage Channel (INS,P1,P2)=(‘70’,’40’,’00’) the initial authentication state is set to “not authenticated” (i.e. *globalPasswordList*, *dfSpecificPasswordList*, *globalSecurityList*, *dfSpecificSecurityList* and *keyReferenceList* are empty, *SessionkeyContext.flagSessionEnabled=noSK*).

(2) If the command SELECT is executed and the *newFile* is an folder the initial authentication state of the selected folder inherit the authentication state of the folder above up the root.

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

(1) The authentication state is changed to "authenticated Human User" for the specific context when the Human User has successfully authenticated via one of the following procedures:

a) VERIFY command using the context specific password or the context specific Multi-Reference password,

b) If the security attribute *flagEnabled* of password object is set to *False* the authentication state for this specific password is changed to "authenticated Human User".

c) If the security attribute *flagEnabled* of Multi-Reference password object is set to *False* the authentication state for this specific Multi-Reference password is changed to "authenticated Human User".

(2) The authentication state is changed to "authenticated Device" for the specific authentication context when a Device has successfully authenticated via one of the following procedures:

c) EXTERNAL AUTHENTICATE with symmetric or public keys,

d) MUTUAL AUTHENTICATE with symmetric or public keys,

e) GENERAL AUTHENTICATE with mutual ELC authentication and

f) GENERAL AUTHENTICATE for asynchronous secure messaging

(3) The effective access rights gained by ECC based CVC: the CHAT are the intersection of the access rights encoded in the CHAT of the CVC chain used as authentication reference data of the Device.

(4) All authentication contexts are lost and the authentication state is set to "not authenticated" for all contexts if the TOE is reset.

(5) If a DELETE command is executed for apassword object or symmetric authentication key the entity is authenticated for the authentication state has to be set to "not authenticated". If a DELETE command is executed for a folder (a) authentication states gained by password objects in the deleted folder shall be set to "not authenticated" and (b) all entires in *keyReferenceList* and *allPublicKeyList* related to the deleted folder shall be removed.

(6) If an authentication attempt using one of the following commands failed the authentication state for the specific context has to be set to "not authenticated": EXTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE, MANAGE SECURITY ENVIRONMENT (variant with restore).

(7) If a context change by using the SELECT command is performed the authentication state for all objects of the old authentication context not belonging to the new context of the performed SELECT command have to be set to "not authenticated".

(8) If failure of secure messaging (not indicated in CLA-byte, or erroneous MAC, or erroneous cryptogram) is detected the authentication status of the device in the current context set to "not authenticated" (i.e. the element in *globalSecurityList* respective in *dfSpecificSecurityList* and the used SK4SM are deleted).

(10)None

7.2.6 Access Control

The TOE shall meet the requirement “Subset access control (FDP_ACC.1/MF_DF)” as specified below.

FDP_ACC.1/MF_DF Subset access control

Hierarchical to : No other components

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/ MF_DF The TSF shall enforce the access control MF_DF SFP on

(1) the subjects *logical channel* bind to users
   a. World,
   b. Human User,
   c. Device,
   d. Human User and Device, none

(2) the objects
   a. all executable code implemented by the TOE,
   b. MF,
   c. Application,
   d. Dedicated file,
   e. Application dedicated file,
   f. Persistent stored public keys
   g. None,

(3) the operation by command following
   a. command SELECT,
   b. create objects with command LOAD APPLICATION with and without command chaining,
   c. delete objects with command DELETE,
   d. read fingerprint with command FINGERPRINT,
   e. command LIST PUBLIC KEY
   f. command GET DATA.

The TOE shall meet the requirement “Security attribute based access control (FDP_ACF.1/ MF_DF)” as specified below.

FDP_ACF.1/ MF_DF 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/ MF_DF

The TSF shall enforce the access control MF_DF SFPto objects based on the following

(1) the subject *logical channel* with security attributes
   a. *interface,*
   b. *globalPasswordList,*
   c. *globalSecurityList,*
   d. *dfSpecificPasswordList,*
   e. *dfSpecificSecurityList,*
   f. *bitSecurityList,*
   g. *SessionkeyContext,*
   h. none

(2) the objects
   a. all executable code implemented by the TOE,
   b. MF with security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules,*
   c. DF with security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules,*
   d. Application with security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules,*
   e. Application dedicated file with security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules,*
   f. Persistent stored public keys
   g. none

FDP_ACF.1.2/ MF_DF

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) SELECT is ***allowed*** if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList,* and *SessionkeyContext* of the subject meet the access rules for the command SELECT of the folder that is to be selected, dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(2) GET CHALLENGE in MF is ***allowed*** if the security attributes *interface, globalPasswordList, globalSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command GET CHALLENGE of the MF dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules es]*].

(3) A subject is allowed to create new objects (user data or TSF data) in the current folder MF if the security attributes *interface, globalPasswordList, globalSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command LOAD APPLICATION of the MF dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(4) A subject is allowed to create new objects (user data or TSF data) in the current folder Application, Dedicated file or Application Dedicated file if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command LOAD APPLICATION of this object dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(5) A subject is allowed to DELETE objects in the current folder MF if the security attributes *interface, globalPasswordList, globalSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command DELETE of the MF dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(6) A subject is allowed to DELETE objects in the current Application, Dedicated file or Application, Dedicated file if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command DELETE of this object dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(7) A subject is allowed to read fingerprint according to FPT_ITE.1 if it is allowed to execute the command FINGERPRINT

(8) All subjects are allowed to execute command LIST PUBLIC KEY to export all persistent stored public keys.

(9) None

FDP_ACF.1.3/ MF_DF The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: None

FDP_ACF.1.4/ MF_DF The TSF shall explicitly deny access of subjects to objects based on the following additional rules:-none.

The TOE shall meet the requirement "Subset access control (FDP_ACC.1/EF)" as specified below.

FDP_ACC.1/EF Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/EF The TSF shall enforce the access control EF SFP on

(1) the subjects *logical channel* bind to users
  a. World,
  b. Human User,
  c. Device,
  d. Human User and Device,
  e. None

(2) the objects
  a. EF,
  b. Transparent EF,
  c. Structured EF,
  d. None

(3) the operation by command following
  a. SELECT,
  b. DELETE of the current file,
  c. None

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1/EF)" as specified below.

FDP_ACF.1/EF Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT MSA.3 Static attribute

FDP_ACF.1.1/EF The TSF shall enforce the access rule EF SFP to objects based on the following

(1) the subject *logical channel* with security attributes

- a. *interface,*
- b. *globalPasswordList,*
- c. *globalSecurityList,*
- d. *dfSpecificPasswordList,*
- e. *dfSpecificSecurityList,*
- f. *bitSecurityList,*
- g. *SessionkeyContext,*
- h. None

(2) the objects

- a. EF with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the EF, and ***transaction protection Mode*** and ***checksum,***
- b. None

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

(1) SELECT is *allowed* if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList*, and *SessionkeyContext* of the subject meet the access rules for the command SELECT of the EF that is to be selected, dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules.*

(2) A subject is allowed to DELETE the current EF if the security attributes *interface, dfSpecificPasswordList, globalSecurityList,globalPasswordList, dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command DELETE of this object dependent on *lifeCycleStatus, interfaceDependentAccessRules* and *seIdentifier* of the current folder.

(3) None

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

FDP_ACF.1.4/EF The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none

The TOE shall meet the requirement “Subset access control (FDP_ACC.1/TEF)” as specified below.

FDP_ACC.1/TEF Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/TEF The TSF shall enforce the access rule TEF SFP on

(1) the subjects *logical channel* bind to users
  a. World,
  b. Human User,
  ~~c. Device,~~
  d. Human User and Device,
  e. None

(2) the objects
  a. Transparent EF,
  b. None

(3) the operation by the following command
  a. ERASE BINARY,
  b. READ BINARY,
  c. SET LOGICAL EOF,
  d. UPDATE BINARY,
  e. WRITE BINARY,
  f. None

The TOE shall meet the requirement “Security attribute based access control (FDP_ACF.1/TEF)” as specified below.

FDP_ACF.1/TEF Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT MSA.3 Static attribute

FDP_ACF.1.1/TEF The TSF shall enforce the access rule TEF SFP to objects based on the following

(1) the subjects *logical channel* with security attributes
  a. *interface,*
  b. *globalPasswordList,*
  c. *globalSecurityList,*
  d. *dfSpecificPasswordList,*
  e. *dfSpecificSecurityList,*
  f. *bitSecurityList,*
  g. *SessionkeyContext,*
  h. None

(2) the objects
  a. with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the current Transparent EF, and ***transaction protection Mode and, checksum,***
  b. None

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

(1) The subject is allowed to execute the command listed in FDP_ACC.1.1/TEF for the current Transparent EF if the security attributes *interface*, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules of this object for this command dependent on *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the current Transparent EF.

(2) none

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

FDP_ACF.1.4/TEF The TSF shall explicitly deny access of subjects to objects based on the following additional rules: Rules defined in FDP_ACF.1.4/EF apply, and none.

Application note: If the checksum of the data to be read by READ BINARY is malicious the TOE must append a warning when exporting. Exporting of malicious data should be taken into account by the evaluator during evaluation of class AVA: vulnerability assessment.

The TOE shall meet the requirement "Subset access control (FDP_ACC.1/SEF)" as specified below.

FDP_ACC.1/SEF	Subset access control
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/ SEF The TSF shall enforce the access rule SEF SFP on

(1) the subjects *logical channel* bind to users
  a. World,
  b. Human User
  c. Device
  d. Human User and Device,
  e. None
(2) the objects
  a. record in Structured EF
  b. None
(3) the operation by command following
  a. APPEND RECORD,
  b. ERASE RECORD,
  c. DELETE RECORD,
  d. READ RECORD,
  e. SEARCH RECORD,
  f. UPDATE RECORD,
  g. None.

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1/SEF)" as specified below.

FDP_ACF.1/SEF	Security attribute based access control
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT MSA.3 Static attribute

FDP_ACF.1.1/SEF The TSF shall enforce the access rule SEF SFP to objects based on the following

(1) the subjects *logical channel* with security attributes
   a. *interface,*
   b. *globalPasswordList,*
   c. *globalSecurityList,*
   d. *dfSpecificPasswordList,*
   e. *dfSpecificSecurityList,*
   f. *bitSecurityList,*
   g. *SessionkeyContext,*
   a. None

(2) the objects
   a. with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the current Structured EF, and *lifeCycleStatus* of the record,
   b. None

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

(1) The subject is allowed to execute the command listed in FDP_ACC.1.1/SEF for the record of the current Structered EF if the security attributes *interface*, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules of this object for this command dependent on *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the current Structered EF, and *lifeCycleStatus* of the record.

(2) None

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

FDP_ACF.1.4/SEF The TSF shall explicitly deny access of subjects to objects based on the following additional rules: Rules defined in FDP_ACF.1.4/EF apply, and None

Application note: If the checksum of the data to be read by READ RECORD is malicious the TOE must append a warning when exporting. Exporting of malicious data should be taken into account by the evaluator during evaluation of class AVA: vulnerability assessment.

The TOE shall meet the requirement "Subset access control (FDP_ACC.1/KEY)" as specified below.

FDP_ACC.1/KEY Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/KEY The TSF shall enforce the access control key SFP on

(1) the subjects *logical channel* bind to users
  a. World,
  b. Human User
  c. Device
  d. Human User and Device,
  e. None

(2) the objects
  a. symmetric key used for user data,
  b. private asymmetric key used for user data,
  c. public asymmetric key for signature verification used for user data,
  d. public asymmetric key for encryption used for user data,
  e. ephemeral keys used during Diffie-Hellmann key exchange,
  f. None

(3) the operation by command following
  a. DELETE for private, public and symmetric key objects,
  b. MANAGE SECURITY ENVIRONMENT,
  c. GENERATE ASYMMETRIC KEY PAIR,
  d. PSO COMPUTE DIGITAL SIGNATURE,
  e. PSO VERIFY DIGITAL SIGNATURE
  f. PSO VERIFY CERTIFICATE,
  g. PSO ENCIPHER,
  h. PSO DECIPHER,
  i. PSO TRANSCIPHER,
  ~~j. PSO COMPUTE CRYPTOGRAPHIC CHECKSUM if supported by the TOE ,~~
  ~~k. PSO VERIFY CRYPTOGRAPHIC CHECKSUM if supported by the TOE,~~ ~~[17]~~
  l. None

[17] PSO COMPUTE/VERIFY CRYPTOGHAPHIC CHECKSUM suppressed compared to PP, because not supported.

The TOE shall meet the requirement“Security attribute based access control (FDP_ACF.1/KEY)” as specified below.

FDP_ACF.1/KEY Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT MSA.3 Static attribute

FDP_ACF.1.1/KEY The TSF shall enforce the access control key SFP to objects based on the following

(1) the subjects *logical channel* with security attributes

a. *interface,*

b. *globalPasswordList,*

c. *globalSecurityList,*

d. *dfSpecificPasswordList,*

e. *dfSpecificSecurityList,*

f. *bitSecurityList,*

g. *SessionkeyContext,*

h. None

(2) the objects

a. symmetric key used for user data with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules*, the *key type* (encryption key or mac key), *interfaceDependentAccessRules* for session keys

b. private asymmetric key used for user data with security attributes *seIdentifier* of the current folder, *lifeCycleStatus, keyAvailable* and *interfaceDependentAccessRules,*

public asymmetric key for signature verification used for user data with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules,*

d. public asymmetric key for encryption used for user data with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules,*

e. CVC with security attributes *certificate content* and *signature*

f. ephemeral keys used during Diffie-Hellmann key exchange

g. None

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

(1) MANAGE SECURITY ENVIRONMENT is ***allowed*** if the security attributes *interface*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the current folder meet the access rules for the command MANAGE SECURITY ENVIRONMENT for the current folder dependent on *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules*, in cases defined in FDP_ACF.1.4/KEY.

(2) A subject is allowed to DELETE an object listed in FDP_ACF.1.1/KEY if the security attributes *interface*, globalPasswordList, globalSecurityList, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command DELETE of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules*,

(3) A subject is allowed to generate a new asymmetric key pair or change the content of existing objects if the security attributes *interface*, globalPasswordList, globalSecurityList, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command GENERATE ASYMMETRIC KEY PAIR of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, *key type* and *interfaceDependentAccessRules*. In case P1='80' or P1='84 the security attribute *keyAvailable* must be set to FALSE.

(4) A subject is allowed to import a public key as part of a CVC by means of the command PSO VERIFY CERTIFICATE if

a) the security attributes *interface*, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO VERIFY CERTIFICATE of the signature public key to be used for verification of the signature of the CVC dependent on *seIdentifier* of the *current folder*, *lifeCycleStatus*, *key type* and *interfaceDependentAccessRules*,

b) the CVC has valid certificate content and signature where the *expiration date* is checked against *pointInTime*.

.

(5) A subject is allowed to compute digital signatures using the private asymmetric key for user data if the security attributes interface, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO COMPUTE DIGITAL SIGNATURE of this object dependent on seIdentifier of the current folder, lifeCycleStatus, the key type and interfaceDependentAccessRules.

(6) Any subject is allowed to verify digital signatures using the public asymmetric key for user data the command PSO VERIFY DIGITAL SIGNATURE

(7) A subject is allowed encrypt user data using the asymmetric key if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO ENCIPHER of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, the *key type* and *interfaceDependentAccessRules*.

(8) A subject is allowed decrypt user data using the asymmetric key if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList,*

dfSpecificSecurityList and *SessionkeyContext* of the subject meet the access rules for the command PSO DECIPHER of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, the *key type* and *interfaceDependentAccessRules*.
(9) A subject is allowed decrypt and to encrypt user data using the asymmetric keys if the security attributes *interface*, *dfSpecificPasswordList*, *globalPasswordList*, *globalSecurityList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO TRANSCIPHER of both keys dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, the *key type* and *interfaceDependentAccessRules*.
~~(10) If the command PSO COMPUTE CRYPTOGRAPHIC CHECKSUM is supported by the TSF than the following rule applies: a subject is allowed to compute a cryptographic checksum with a symmetric key used for user data if the security attributes *interface*, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO COMPUTE CRYPTOGRAPHIC CHECKSUM of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, the *key type* and *interfaceDependentAccessRules*.~~
~~(11) If the command PSO VERIFY CRYPTOGRAPHIC CHECKSUM is supported by the TSF than the following rule applies: a subject is allowed to verify a cryptographic checksum with a symmetric key used for user data if the security attributes *interface*, *globalPasswordList*, *globalSecurityList*, *dfSpecificPasswordList*, *dfSpecificSecurityList* and *SessionkeyContext* of the subject meet the access rules for the command PSO VERIFY CRYPTOGRAPHIC CHECKSUM of this object dependent on *seIdentifier* of the current folder, *lifeCycleStatus*, the *key type* and *interfaceDependentAccessRules*.~~[18]
(12) None

[18] PSO COMPUTE/VERIFY CRYPTOGRAPHIC CHECKSUM suppressed compared to PP, because not supported.

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

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

(1) If the security attribute *keyAvailable=TRUE* the TSF shall prevent generation of a private key by means of the command GENERATE ASYMMETRIC KEY PAIR with P1='80' or P1='84.

(2) None.

The TOE shall meet the requirement "Subset access control (FDP_ACC.1/PKEYS)" as specified below.

FDP_ACC.1/PKEYS Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/ PKEYS The TSF shall enforce the PKEYS SFP on Initializer, PersoKeys, and Import of Perso Keys.

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1/PKEYS)" as specified below.

FDP_ACF.1/PKEYS Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT MSA.3 Static attribute

FDP_ACF.1.1/ PKEYS The TSF shall enforce the access rule PKEYS SFP to objects based on the following:

Subject Inizializer with his corresponding authentication state and object Perso Keys

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

The Initializer is allowed to import Perso Keys only if his authentication state is present.

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

FDP_ACF.1.4/ PKEYS The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none.

The TOE shall meet the requirement “Import of user data without security attributes (FDP_ITC.1/PKEYS)” as specified below.

FDP_ITC.1/PKEYS	Subset access control
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 initialisation

FDP_ITC.1.1/ PKEYS — The TSF shall enforce the PKEYS SFP when importing user data, controlled under the SFP, from outside of the TOE..

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

FDP_ITC.1.3/ PKEYS — The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: none.

The TOE shall meet the requirement “Specification of Management Functions (FMT_SMF.1)” as specified below.

FMT_SMF.1	Specification of 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) Initialization,by means of commands LoadSecret,WriteImage and VerifyImage,
(2) Personalization by means of command StoreData,
(3) Life Cycle Management by means of commands GENERATE ASYMMETRIC KEY PAIR, DELETE, LOAD APPLICATION, TERMINATE, TERMINATE DF, TERMINATE CARD USAGE, None
(4) Management of access control security attributes by means of commands ACTIVATE, DEACTIVATE, ACTIVATE RECORD, DEACTIVATE RECORD, ENABLE VERIFICATION REQUIREMENT, DISABLE VERIFICATION REQUIREMENT, LOAD APPLICATION,
(5) Management of password objects attributes by means of commands CHANGE REFERENCE DATA, RESET RETRY COUNTER, GET PIN STATUS, VERIFY, LOAD APPLICATION,
(6) Management of device authentication reference data by means of commands PSO VERIFY CERTIFICATE, GET SECURITY STATUS KEY, LOAD APPLICATION
(7) None

Application Note: The modifications compared to the PP in points (1) and (2) are no functional modification but just the concrete listing of the relevant commands. Therefore no other SFR is undermined.

The TOE shall meet the requirement “Management of security attributes (FMT_MSA.1/Life)” as specified below.

FMT_MSA.1/Life	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/Life The TSF shall enforce the access control MF_DF SFP, access control EF SFP, access rule TEF SFP, access rule SEF SFP and access control key SFP to restrict the ability to

(1) ***create*** **all** security attributes of the new object DF, Application, Application dedicated file, EF, TEF and SEF to subjects allowed execution of command LOAD APPLICATION for the MF, DF, Application, Application dedicated file where the new object is created,

(2) ***change*** **the security attributes** of the object MF, DF, Application, Application dedicated file, EF, TEF and SEF by means of command LOAD APPLICATION to ***subjects allowed execution of command LOAD APPLICATION for theMF, DF, Application, Application dedicated file where the object is updated***

(3) ***change*** **the security attributes** ***lifeCycleStatus*** **to„*Operational state (active)*“ to subjects allowed execution of command ACTIVATE for the selected object,**

(4) ***change*** **the security attributes** ***lifeCycleStatus*** **to „*Operational state (deactivated)*” to subjects allowed execution of command DEACTIVATE for the selected object,**

(5) ***change*** **the security attributes** ***lifeCycleStatus*** **to„*Termination state*” to subjects allowed execution of command TERMINATE for the selected EF , the key object or the password object,**

(6) ***change*** **the security attributes *lifeCycleStatus* to „*Termination state*” to subjects allowed execution of command TERMINATE DF for the selected DF , Application or Application File**

(7) ***change*** **the security attributes *lifeCycleStatus* to “*Termination state*” to subjects allowed execution of command TERMINATE CARD USAGE,**

(8) ***query*** **the security attributes *lifeCycleStatu* to by means of command SELECT to subjects *allowed* execution of command SELECT for the object to be selected,**

(9) ***delete*** **all security attributes of the selected object to subjects allowed execution of command DELETE for the selected object to none**

The subject *logical channel* is allowed to execute a command if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList, bitSecurityList SessionkeyContext* of the subject meet the security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules* of the affected object.

The TOE shall meet the requirement "Management of security attributes (FMT_MSA.1/SEF)" as specified below.

FMT_MSA.1/SEF 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/SEF The TSF shall enforce the access control SEF SFP to restrict the ability to

(1) ***change*** the security attributes *lifeCycleStatus* of the selected record to „*Operational state (active)*" to subjects allowed to execute the command ACTIVATE RECORD

(2) ***change*** **the security attributes *lifeCycleStatus* of the selected record to „*Operational state (deactivated)*" to subjects allowed to execute the command DEACTIVATE RECORD,**

(3) ***delete*** **all security attributes of the selected record to subjects allowed to execute the command DELETE RECORD,**

(4) **None.**

The subject *logical channel* is allowed to execute a command if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList, bitSecurityList SessionkeyContext* of the subject meet the security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules* of the affected object.

The TOE shall meet the requirement “Static attribute initialisation (FMT_MSA.3)” as specified below

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 the access control MF_DF SFP, access control EF SFP, access rule TEF SFP, access rule SEF SFP, access control key SFP and PKEYS_SFP to provide ***restrictive*** default values for security attributes that are used to enforce the SFP. **After reset the security attributes of the subject are set as follows**

- *(1)* **currentFolder is root,**
- *(2)* **keyReferenceList, globalSecurityList, globalPasswordList, dfSpecificSecurityList, dfSpecificPasswordList and bitSecurityList are empty,**
- *(3)* **SessionkeyContext.flagSessionEnabled is set to noSK,**
- **(4) *seIdentifier* is #1,**
- ***(5) currentFile is undefined,***
- **(6) *Authentication state of Initializer is not present (only relevant during production phases)***

FMT_MSA.3.2 *The TSF* shall allow the subjects allowed to execute the command LOAD APPLICATION to specify alternative initial values to override the default values when an object or information is created.

The TOE shall meet the requirement “Management of TSF data - PIN (FMT_MTD.1/PIN)” as specified below.

FMT_MTD.1/PIN Management of TSF data - PIN

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/PIN The TSF shall restrict the ability to

(1) ***set*** new *secret* of the password objects by means of command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00) to subjects successful authenticated with the old *secret* of this password object,

(2) ***set* new *secret* and change *transportStatus* to regularPassword of the password objects with *transportStatus* equal to Leer-PIN to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01]**

(3) ***set* new *secret* of the password objects by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,00) to subjects successful authenticated with the PUC of this password object**

(4) ***set* new *secret* of the password objects by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02) to subjects allowed to execute the command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02)**

.

Application note: The commands CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01) and RESET RETRY COUNTER (CLA,INS,P1)=(00,2C,02) set a new password without need of authentication by PIN or PUC. In order to prevent bypass of the human user authentication defined by the PIN or PUC the object system shall define access control to this command as required by the security needs of the specific application context, cf. OE.Resp-ObjS.

The TOE shall meet the requirement "Management of security attributes - PIN (FMT_MSA.1/PIN)" as specified below.

FMT_MSA.1/PIN	Management of security attributes - PIN
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/PIN The TSF shall enforce the access control MF_DF SFP, access control EF SFP, access rule TEF SFP, access rule SEF SFP and access control key SFP to restrict the ability to

(1) *reset* **by means of commands VERIFY the security attribute retry counter of password objects to subjects successful authenticated with the secret of this password object,**

(2) *reset* **by means of commands CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00) the security attribute retry counter of password objects to subjects successful authenticated with the old secret of this password object ,**

(3) *change* **by means of commands CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00) the security attribute *transportStatus* from Transport-PIN to regularPassword to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00),**

(4) *change* **by means of commands CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01) the security attribute *transportStatus* from Leer-PIN to regularPassword to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01),**

(5) reset by means of commands DISABLE VERIFICATION REQUIREMENT with (CLA,INS,P1)=(00,26,00) the security attribute retry counter of password objects to subjects successful authenticated with the old secret of this password object ,

(6) ***reset* by means of commands ENABLE VERIFICATION REQUIREMENT with (CLA,INS,P1)=(00,28,00) the security attribute retry counter of password objects to subjects successful authenticated with the old secret of this password object ,**

(7) ***reset* by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,00) or (CLA,INS,P1)=(00,2C,01) the security attribute retry counter of password o bjects to subjects successful authenticated with the PUC of this password objet**

(8) ***reset* by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02) or (CLA,INS,P1)=(00,2C,03) the security attribute retry counter of password objects to subjects allowed to execute the command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02) or (CLA,INS,P1)=(00,2C,03)**

(9) query by means of command GET PIN STATUS the secur ity attribute flagEnabled, retry counter, transportStatus to World

(10) ***enable* the security attributes *flagEnabled* requiri ng authentication *with the selected password to subjects* authenticated *with* pas swaord and allowed *to execute* the comman d ENABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,28 ,00)**

(11) ***enable* the security attributes *fla gEnabled* requiring authentication with the selected password to subjects allowed to execute the command ENABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,28 ,01)**

(12) ***disable* the security attributes *flagEnabled* requiring authentication with the selected password to subjects authenticated with passwaord and allowed to execute the command DISABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,26,00)**

(13) ***disable* the security attributes *flagEnabled* requiring authentication with the selected password to subjects allowed to execute the command DISABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,26,01)**

Application note: The commands RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02) or (CLA,INS,P1)=(00,2C,03) allows to reset the RESET RETRY COUNTER without authentication with PUC. In order to prevent bypass of the human user authentication defined by the PIN the object system shall define access control to these commands as required by the security needs of the specific application context, cf. OE.Resp-ObjS.

The TOE shall meet the requirement "Management of TSF data – Authentication data (FMT_MTD.1/Auth)" as specified below.

FMT_MTD.1/Auth	Management of TSF data – Authentication data
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/ Auth	The TSF shall restrict the ability to (1) *import by means of commands LOAD* APPLICATION the root public keys to roles autorized to execute this command, **(2) *import by means of commands PSO VERIFY CERTIFICATE* the root public keys to roles autorized to execute this command,** **(3) *import by means of commands PSO VERIFY CERTIFICATE* the certificates as device authentication reference data to roles autorized to execute this command**, **(4) *select by means of command MANAGE SECURITY ENVIRONMENT* the device authentication reference data to *roles autorized to execute this command.*** **The subject *logical channel* is allowed to execute a command if the security attributes *interface, globalPasswordList, globalSecurityList, dfSpecificPasswordList, dfSpecificSecurityList* and *bitSecurityList SessionkeyContext* of the subject meet the security attributes *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules* of the affected object.**

The TOE shall meet the requirement “Management of security attributes (FMT_MSA.1/Auth)” as specified below.

FMT_MSA.1/Aut	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/ Auth	The TSF shall enforce the access control key SFP to restrict the ability to ***query*** the security attributes access control rights set for the key to meet the access rules of command GET SECURITY STATUS KEY of the object dependent on *lifeCycleStatus, seIdentifier* and *interfaceDependentAccessRules* .

The TOE shall meet the requirement “Management of TSF data – No export (FMT_MTD.1/NE)” as specified below.

FMT_MTD.1/NE	Management of TSF data – No export
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management

FMT_MTD.1.1/NE The TSF shall restrict the ability to

(1) ***export*** TSF data according to FPT_ITE.2the

(a) public authentication reference data,

(b) security attributes for objects of the object system to all subjects allowed to execute GET OBJ INFO and SELECT

(2) ***export*** **TSF data according to FPT_ITE.2 the no other security attributes to all subjects allowed to execute GET OBJ INFO**

(3) ***export*** **the following TSF-data**

(a) Password

(b) Multi-Reference password

(c) PUC

(d) Private keys

(e) Session keys

(f) Symmetric authentication keys

(g) Private authentication keys

(h) Kicc

(i) Persos keys

(j) K_Verify

(k) None

and the following user data

(l) Private keys of the user

(m) Symmetric keys of the user

(n) None

to nobody

The TOE shall meet the requirement "Management of TSF data – TOE Image (FMT_MTD.1/Init)" as specified below.

FMT_MTD.1/Init Management of TSF data – TOE Image

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/ Init The TSF shall restrict the ability to load the Product Image to the authenticated Initializer.

Implementation note: The Initializer authenticates with the card individual K_{ICC} to be able to load the Product Image secured with the K_Verify by the developper. The TSF accept this image only after successfully verifying this developer's signature contained within the image data.

This is an additional SFR compared to the PP. It describes the authentication functionality for the Initialization phase, which is totally separated from the usage phase, where the SFRs from the PP apply. Moreover, the gained authentication state does not allow loading other images (i.e. data structures and filter code) than those digitally signed by the developer, and no sensitive data can be read out from the TOE. Therefore no other SFR is undermined. This also holds for the extension of the dependent FMT_SMR.1.

The TOE shall meet the requirement “Management of TSF data – Personalization Data (FMT_MTD.1/Perso)” as specified below.

FMT_MTD.1/Perso	Management of TSF data – Personalization Data
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/ Perso	The TSF shall restrict the ability to import Personalization Data to the authenticated Personalizer.

Implementation note: The Personalizer authenticates with the card individual Perso Keys and derives session keys from them to be able to import Personalization Data in CPS scheme. Those data are MAC protected and encrypted according to their sensitivity. “Personalization Data” are that part of the assets (user data in EF, secret keys, private keys, public keys) that will be established via the personalization process.

This is an additional SFR compared to the PP. It describes the authentication functionality for the Personalization phase, which is totally separated from the usage phase, where the SFRs from the PP apply. Moreover, the gained authentication state does only allow loading of personalization data, which are not part of the TOE. Also it is not possible to read out sensitive data from the TOE. Therefore no other SFR is undermined. This also holds for the extension of the dependent FMT_SMR.1.

7.2.7 Cryptographic Functions

The TOE provides cryptographic services based on elliptic curve cryptography (ECC) using the following curves refered to as COS standard curves in the following

(1) length 256 bit
   (a) brainpoolP256r1 defined in RFC5639 [41],
   (b) ansix9p256r1 defined in ANSI X.9.62 [42],
(2) length 384 bit
   (a) brainpoolP384r1 defined in RFC5639 [41],
   (b) ansix9p384r1 defined in ANSI X.9.62 [42],
(3) length 512 bit
   (a) brainpoolP512r1] defined in RFC5639 [41].

The Authentication Protocols produce agreed parameters to generate the message authentication key and – if secure messaging with encryption is required - the encryption key for secure messaging. Key agreement for *rsaSessionkey4SM* uses RSA only with 2048 bitmodulo length.

The TOE shall meet the requirement "Random number generation (FCS_RNG.1)" as specified below.

FCS_RNG.1	Random number generation
Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_RNG.1.1 The TSF shall provide ***a physical*** random number generator ***PTG.2*** that implements:

- PTG.2.1 A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
- PTG.2.2 If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that
- depends on some raw random numbers that have been generated after the total failure of the entropy source.
- PTG.2.3 The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.
- PTG.2.4 The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.
- PTG.2.5 The online test procedure checks the quality of the raw random number sequence. It is triggered continuously. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

FCS_RNG.1.2 The TSF shall provide random numbers that meet :

- PTG.2.6 Test procedure A, as defined in [5] does not distinguish the internal random numbers from output sequences of an ideal RNG.
- PTG.2.7 The average Shannon entropy per internal random bit exceeds 0.997.

The TOE shall meet the requirement "Cryptographic operation - SHA (FCS_COP.1/SHA)" as specified below.

FCS_COP.1/SHA	Cryptographic operation - SHA
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/SHA	The TSF shall perform hashing in accordance with a specified cryptographic algorithm (1) SHA-1, (2) SHA-256, (3) SHA-384, (4) SHA-512 and cryptographic key sizes none that meet the following TR-03116, FIPS 180-4

The TOE shall meet the requirement "Cryptographic key generation – 3TDES_SM (FCS_CKM.1/3TDES_SM)" as specified below.

FCS_CKM.1/ 3TDES_SM	Cryptographic key generation – 3TDES_SM
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/ 3TDES_SM	The TSF shall generate session cryptographic keys in accordance with a specified cryptographic key generation algorithm Key Derivation Function specified in sec. 5.6.3 in ANSI X9.63 and specified cryptographic key sizes 192 bit (168 bit effectively) that meet the following: ANSI X9.63 .

The TOE shall meet the requirement “Cryptographic operation - COS for 3TDES (FCS_COP.1/COS.3TDES)” as specified below.

FCS_COP.1/ COS.3TDES	Cryptographic operation - COS for 3TDES
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/ COS.3TDES

The TSF shall perform decryption and encryption for secure messaging in accordance with a specified cryptographic algorithm 3TDES in CBC mode and cryptographic key size 192 bit (168 bit effectively) that meet the following TR-03116 , NIST SP 800-67

The TOE shall meet the requirement “Cryptographic operation COS for RMAC (FCS_COP.1/COS.RMAC)” as specified below

FCS_COP.1/COS.RMAC	Cryptographic operation COS for RMAC
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/ COS.RMAC

The TSF shall perform

(1) computation and verification of cryptographic checksum for command
    a. MUTUAL AUTHENTICATE,
    b. EXTERNAL AUTHENTICATE,
(2) computation and verification of cryptographic checksum for secure messaging

in accordance with a specified cryptographic algorithm Retail MAC and cryptographic key sizes 192 bit (168 bit effectively) that meet the following TR-03116 , COS specification

The TOE shall meet the requirement “Cryptographic operation – COS for AES (FCS_COP.1/COS.AES)” as specified below.

FCS_COP.1/ COS.AES	Cryptographic operation – COS for AES
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/ COS.AES	The TSF shall perform 1. encryption and decryption with card internal key for commands a. MUTUAL AUTHENTICATE, b. EXTERNAL AUTHENTICATE, 2. encryption with card internal key for command INTERNAL AUTHENTICATE, 3. encryption and decryption with card internal key for command GENERAL AUTHENTICATE, 4. encryption and decryption for secure messaging in accordance with a specified cryptographic algorithm AES in CBC modeand cryptographic key sizes 128 bit, 192 bit, 256 bit that meet the following: TR-03116 , COS specification , FIPS 197 .

The TOE shall meet the requirement “Cryptographic key generation – COS for SM keys (FCS_CKM.1/AES.SM)” as specified below.

FCS_CKM.1/ AES.SM	Cryptographic key generation – COS for SM keys
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/ AES.SM	The TSF shall generate **session** cryptographic keys in accordance with a specified cryptographic key generation algorithm Key Derivation for AES as specified in sec. 4.3.3 in TR-03111 and specified cryptographic key sizes 128 bit, 192 bit, 256 bit that meet the following TR-03111, COS specification FIPS 197 .

The TOE shall meet the requirement "Cryptographic operation – COS for CMAC (FCS_COP.1/COS.CMAC)" as specified below.

FCS_COP.1/ COS.CMAC Cryptographic operation – COS for CMAC

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/ COS.CMAC The TSF shall perform

(1) computation and verification of cryptographic checksum for command
   a. MUTUAL AUTHENTICATE,
   b. EXTERNAL AUTHENTICATE,
(2) computation of cryptographic checksum for command INTERNAL AUTHENTICATE,
(3) computation and verification of cryptographic checksum for secure messaging

in accordance with a specified cryptographic algorithm CMAC and cryptographic key sizes 128 bit, 192 bit, and 256 bit that meet the following TR-03116 , COS specification NIST SP 800-38B.

The TOE shall meet the requirement "Cryptographic key generation – RSA key generation (FCS_CKM.1/RSA)" as specified below.

FCS_CKM.1/RSA Cryptographic key generation – RSA 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/RSA The TSF shall generate cryptographic **RSA** keys in accordance with a specified cryptographic key generation algorithm GEMALTO Proprietary Algorithm and specified cryptographic key 2048 bit and 3072 bit modulo length that meet the following TR- 03116 .

The TOE shall meet the requirement "Cryptographic key generation – ECC key generation (FCS_CKM.1/ELC)" as specified below.

FCS_CKM.1/ELC	Cryptographic key generation – ECC 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/ELC	The TSF shall generate cryptographic **ELC** keys in accordance with a specified cryptographic key generation algorithm GEMALTO Proprietary Algorithm **with COS standard curves** and specified cryptographic key 256 bit, 384 bit and 512 bit that meet the following TR-03111 , COS specification .

The TOE shall meet the requirement "Cryptographic operation – RSA signature-creation (FCS_COP.1/COS.RSA.S)" as specified below.

FCS_COP.1/ COS.RSA.S	Cryptographic operation – RSA signature-creation
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/ COS.RSA.S	The TSF shall perform digital signature generation for commands (1) PSO COMPUTE DIGITAL SIGNATURE, (2) INTERNAL AUTHENTICATE in accordance with a specified cryptographic algorithm (1) RSASSA-PSS-SIGN with SHA-256, (2) RSA SSA PKCS1-V1_5, (3) RSA ISO9796-2 DS1 with SHA-256 (for INTERNAL AUTHENTICATE only), (4) RSA ISO9796-2 DS2 with SHA-256 (for PSO Compute DIGITAL SIGNATURE only) , and cryptographic key sizes (1) 2048 bit modulo length, (2) 3072 bit modulo length that meet the following: TR-03116 , COS specification, ISO/IEC 9796-2, PKCS #1.

The TOE shall meet the requirement "Cryptographic operation – RSA signature verification (FCS_COP.1/COS.RSA.V)" as specified below.

FCS_COP.1/COS.RSA.V	Cryptographic operation – RSA signature verification
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 / COS.RSA.V	The TSF shall perform digital signature verification for import of RSA keys using the commands (1) PSO VERIFY CERTIFICATE, (2) EXTERNAL AUTHENTICATE in accordance with a specified cryptographic algorithm RSA ISO9796-2 DS1 and cryptographic key sizes 2048 bit modulo length that meet the following: TR-03116 , COS specification, , ISO/IEC 9796-2, PKCS #1 .

The TOE shall meet the requirement "Cryptographic operation – ECDSA signature verification (FCS_COP.1/COS.ECDSA.V)" as specified below.

FCS_COP.1/COS.ECDSA.V	Cryptographic operation – ECDSA signature verification
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/ COS.ECDSA.V	The TSF shall perform digital signature verification for import of ELC keys for the commands (1) PSO VERIFY CERTIFICATE, (2) PSO VERIFY DIGITAL SIGNATURE, (3) EXTERNAL AUTHENTICATE in accordance with a specified cryptographic algorithm ECDSA with COS standard curves using (1) SHA-256, (2) SHA-384, (3) SHA-512 and cryptographic key sizes 256 bits, 384 bits, 512 bits that meet the following TR-03111 , TR-03116 , COS specification, X9.63 .

The TOE shall met the requirement "Cryptographic operation – ECDSA signature-creation (FCS_COP.1/COS.ECDSA.S)" as specified below.

FCS_COP.1/COS.ECDSA.S	Cryptographic operation – ECDSA signature-creation
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/ COS.ECDSA.S

The TSF shall perform digital signature generation for command

(1) PSO COMPUTE DIGITAL SIGNATURE,

(2) INTERNAL AUTHENTICATE

in accordance with a specified cryptographic algorithm ECDSA with COS standard curves using

(1) SHA-256,

(2) SHA-384

(3) SHA-512

and cryptographic key sizes 256 bits, 384 bits, 512 bits that meet the following TR-03111 , TR-03116 , COS specification, X9.63

The TOE shall meet the requirement "Cryptographic operation – RSA encryption and (FCS_COP.1/COS.RSA)" as specified below.

FCS_COP.1/COS.RSA	Cryptographic operation – RSA encryption and decryption
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/ COS.RSA

The TSF shall perform

(1) encryption with passed key for command PSO ENCIPHER,
(2) decryption with stored key for command PSO DECIPHER,
(3) decryption and encryption for command PSO TRANSCIPHER using RSA (transcipher of data using RSA keys),
(4) decryption for command PSO TRANSCIPHER using RSA (transcipher of data from RSA to ELC),
(5) encryption for command PSO TRANSCIPHER using RSA (transcipher of data from ELC to RSA)

in accordance with a specified cryptographic algorithm

(6) for encryption:
  a. RSAES-PKCS1-v1_5_Encrypt (PKCS#1 section 7.2.1),
  b. RSA-OAEP-Encrypt (PKCS#1 section 7.1.1]),
(7) for decryption:
  a. RSAES-PKCS1-v1_5_Decrypt (PKCS#1section 7.2.2],
  b. RSA-OAEP-Decrypt (PKCS#1 section 7.1.2])

and cryptographic key sizes 2048 bit and 3072bit modulo length for RSA private key operation, 2048 bit and 3072 bit length for RSA public key operation and 256 bit, 384 bit and 512 bit for the COS standard curves that meet the following TR-03116 , COS specification, PKCS#1

This SFR was refined compared to the PP with 3072 bit modulus length for RSA public key operation. This is using even longer modulus length than originally allowed; therefore no other SFR is undermined.

The TOE shall meet the requirement "Cryptographic operation – ECC encryption and decryption (FCS_COP.1/COS.ELC)" as specified below.

FCS_COP.1/ COS.ELC Cryptographic operation – ECC encryption and decryption

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/ COS.ELC

The TSF shall perform
(1) encryption with passed key for command PSO ENCIPHER,
(2) decryption with stored key for command PSO DECIPHER,
(3) decryption and encryption for command PSO TRANSCIPHER using ELC (transcipher of data using ELC keys),
(4) decryption for command PSO TRANSCIPHER using ELC (transcipher of data from ELC to RSA),
(5) encryption for command PSO TRANSCIPHER using ELC (transcipher of data from RSA to ELC)
in accordance with a specified cryptographic algorithm
(1) for encryption ELC encryption,
(2) for decryption ELC decryption
and cryptographic key sizes 2048 bit and 3072 bit modulo length for RSA private key operation, 2048 bit and 3072 bit length for RSA public keys operation and 256 bits, 384 bits, 512 bits for ELC keys with COS standard curves that meet the following TR-03111 , TR-03116 , and COS specification .

This SFR was amended compared to the PP with 3072 bit modulus length for RSA public key operation. This is using even longer modulus length than originally allowed; therefore no other SFR is undermined.

The TOE shall meet the requirement "Cryptographic key destruction (FCS_CKM.4)" as specified below.

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: the allocated memory is overwritten with' 00…00'
that meets the following: None.

The TOE shall meet the requirement "Cryptographic operation – Production Auth (FCS_COP.1/ **PAUTH**)" as specified below.

FCS_COP.1/ PAUTH	Cryptographic operation – Production Auth
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/ PAUTH	The TSF shall perform authentication in productive phases with decryption and encryption and MAC computation and verification in accordance with a specified cryptographic algorithm 2TDES in CBC and Retail MAC mode and cryptographic key size 128 bit (112 bit effectively) that meet the following [FIPS PUB 46-3].

This is an additional SFR compared to the PP. It describes the authentication functionality for the Initialization and Personalization phase, which are totally separated from the functionality of the usage phase, where the SFRs from the PP apply. Therefore no other SFR is undermined. This also holds for the dependent additional SFIs FDP_ITC.1/PKEYS, FDP_ACC.1/PKEYS, and FDP_ACF.1/PKEYS and the corresponding amendment in FMT_MSA.3.

7.2.8 Protection of communication

The TOE shall meet the requirement "Inter-TSF trusted channel (FTP_ITC.1/TC)" as specified below.

FTP_ITC.1/TC	Inter-TSF trusted channel
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1/TC	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/TC	The TSF shall permit ***another trusted IT product*** to initiate communication via the trusted channel.
FTP_ITC.1.3/TC	The TSF shall initiate communication via the trusted channel for none.

7.3 SECURITY ASSURANCE REQUIREMENTS FOR THE TOE

The TOE security assurance requirements define the assurance requirements for the TOE using only assurance components drawn from [CCPART3].

The assurance requirements are:

Class ADV: Development	
Architectural design	(ADV_ARC.1)
Functional specification	(ADV_FSP.4)
Implementation representation	(ADV_IMP.1)
TOE design	(ADV_TDS.3)
Class AGD: Guidance documents	
Operational user guidance	(AGD_OPE.1)
Preparative user guidance	(AGD_PRE.1)
Class ALC: Life-cycle support	
CM capabilities	(ALC_CMC.4)
CM scope	(ALC_CMS.4)
Delivery	(ALC_DEL.1)
Development security	(ALC_DVS.2)
Life-cycle definition	(ALC_LCD.1)
Tools and techniques	(ALC_TAT.1)
Class ASE: Security Target evaluation	
Conformance claims	(ASE_CCL.1)
Extended components	(ASE_ECD.1)
ST introduction	(ASE_INT.1)
Security objectives	(ASE_OBJ.2)
Derived security requirements	(ASE_REQ.2)
Security problem definition	(ASE_SPD.1)
TOE summary specification	(ASE_TSS.1)
Class ATE: Tests	
Coverage	(ATE_COV.2)
Depth	(ATE_DPT.2)
Functional tests	(ATE_FUN.1)
Independent testing	(ATE_IND.2)
Class AVA: Vulnerability assessment	
Vulnerability analysis	(AVA_VAN.5)

Table 31 –Assurance components

7.3.1 Refinements of the TOE Assurance Requirements

Refinements regarding	Reference to BSI-PP-0035
Delivery procedure (ALC_DEL)	Section 6.2.1.1 “Refinements regarding Delivery procedure (ALC_DEL)”
Development Security (ALC_DVS)	Section 6.2.1.2 “Refinements regarding Development Security (ALC_DVS)”
CM scope (ALC_CMS)	Section 6.2.1.3 “Refinements regarding CM scope (ALC_CMS)”
CM capabilities (ALC_CMC)	Section 6.2.1.4 “Refinements regarding CM capabilities (ALC_CMC)”
Security Architecture (ADV_ARC)	Section 6.2.1.5 “Refinements regarding Security Architecture (ADV_ARC)”
Functional Specification (ADV_FSP)	Section 6.2.1.6 “Refinements regarding Functional Specification (ADV_FSP)”
Implementation Representation (ADV_IMP)	Section 6.2.1.7 “Refinements regarding Implementation Representation (ADV_IMP)”
Test Coverage (ATE_COV)	Section 6.2.1.8” Refinements regarding Test Coverage (ATE_COV)”
User Guidance (AGD_OPE)	Section 6.2.1.9 “Refinements regarding User Guidance (AGD_OPE)”
Preparative User Guidance (AGD_PRE)	Section 6.2.1.10 “Refinements regarding Preparative User Guidance (AGD_PRE)”
Refinement regarding Vulnerability Analysis (AVA_VAN)	Section 6.2.1 “Refinement regarding Vulnerability Analysis (AVA_VAN)”

Table 32 –Refined TOE assurance requirements

The following sections define refinements and application notes to the chosen SAR.

Refinements to ADV_ARC.1 Security architecture description
The ADV_ARC.1 Security architecture description requires as developer action
ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
and the related content and presentation element
ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
The COS specification allows implementation of optional features and commands. The following refinement for ADV_ARC.1.5C defines specific evidence required for these optional features and commands if implemented by the TOE and not being part of the TSF.
Refinement: If a feature or command identified as optional in the COS specification is implemented in the TOE or any other additional functionality of the TOE is not part of the TSF

the security architecture description shall demonstrate that it do not bypass the SFR-enforcing functionality.
This TOE doesn't implement any optional features.

Refinements to ADV_FSP.4 Complete functional specification
The following content and presentation element of ADV_FSP.4 Complete functional specification is refined as follows:

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.
Refinement: The functional specification shall describe the purpose and method of use for all TSFI including
(1) the physical and logical interface of the smart card platform, both contact based and contactless as implemented by the TOE,
(2) the logical interface of the wrapper to the verification tool.
Application note: The IC surface as external interface of the TOE provides the TSFI for physical protection (cf. FPT_PHP.3) and evaluated in the IC evaluation as base evaluation for the composite evaluation of the composite. This interface is also analyzed as attack surface in the vulnerability analysis e.g. in respect to perturbation and emanation side channel analysis.

Refinement to ADV_IMP.1
The following content and presentation element of ADV_IMP.1 Implementation representation of the TSF is refined as follows:

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TOE.
Application note : The refinement extends the TSF implementation representation to the TOE implementation representation, i.e. the complete executable code implemented on the Security platform IC including all IC Embedded Software and especially the Card Operating System, (COS).

Refinements to AGD_OPE.1 Operational user guidance
The following content and presentation element of AGD_OPE.1 Operational user guidance is refined as follows:

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
Refinement: The operational user guidance shall describe the method of use of the wrapper interface.
Application note : The wrapper will be used to interact with the smartcard for export of all public TSF data of all objects in an object system according to "Export of TSF data (FPT_ITE.2)". Because the COS specification identifies optional functionality the TOE may support the guidance documentation shall describe method of use of the TOE (as COS, wrapper) to find all objects in the object system and to export all security attributes of these objects.

Refinements to ATE_FUN.1 Functional tests
The following content and presentation element of ATE_FUN.1 Functional tests is refined as follows:

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

Refinements to ATE_IND.2 Independent testing – sample

The following content and presentation element of ATE_IND.2 Functional tests is refined as follows:

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

.

7.4 SECURITY REQUIREMENTS RATIONALE

7.4.1 Security Functional Requirements Rationale

The following table provides an overview for security functional requirements coverage also giving an evidence for sufficiency and *necessity* of the SFRs chosen.

	O.Identification	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND
FAU_SAS.1/SICP	X							
FCS_RNG.1/SICP								X
FDP_IFC.1/SICP		X				X		
FDP_ITT.1/SICP		X				X		
FMT_LIM.1/SICP							X	
FMT_LIM.2/SICP							X	
FPT_FLS.1/SICP				X				
FPT_ITT.1/SICP		X				X		
FPT_PHP.3/SICP			X		X			
FRU_FLT.2/SICP				X				

Table 33 –Coverage of Security Objectives for the TOE IC part by SFR

The objectives and SFRs as used in previous Table are defined and handled in BSI-CC-PP-0035-2007. Hence, the rationale for these items and their correlation is given in BSI-PP-0035 and not repeated here.

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging
FDP_RIP.1		X							
FDP_SDI.2/Internal	X								
FDP_SDI.2/ReadEF	X								
FPT_FLS.1	X	X							
FPT_EMS.1		X							
FPT_TDC.1				X					
FPT_ITE.1				X					
FPT_ITE.2				X					
FPT_TST.1	X	X	X						
FIA_SOS.1					X				
FIA_AFL.1/PIN					X				
FIA_AFL.1/PUC					X				
FIA_ATD.1					X				
FIA_UAU.1					X				
FIA_UAU.4					X				
FIA_UAU.5					X				
FIA_UAU.6					X				
FIA_UID.1					X				
FIA_API.1					X				
FMT_SMR.1					X	X			
FIA_USB.1					X	X			
FDP_ACC.1/MF_DF						X			
FDP_ACF.1/MF_DF						X			
FDP_ACC.1/EF						X			
FDP_ACF.1/EF						X			
FDP_ACC.1/TEF						X			
FDP_ACF.1/TEF						X			
FDP_ACC.1/SEF						X			
FDP_ACF.1/SEF						X			
FDP_ACC.1/KEY						X	X		
FDP_ACF.1/KEY						X	X		
FDP_ACC.1/PKEYS						X	X		
FDP_ACF.1/PKEYS						X	X		

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging
FDP_ITC.1/PKEYS							X		
FMT_MSA.3						X			
FMT_SMF.1						X			
FMT_MSA.1/Life						X	X		
FMT_MSA.1/SEF						X			
FMT_MTD.1/PIN					X	X			
FMT_MSA.1/PIN					X	X			
FMT_MTD.1/Auth					X	X			
FMT_MSA.1/Auth					X	X			
FMT_MTD.1/NE		X				X			
FMT_MTD.1/Init						X			
FMT_MTD.1/Perso						X			
FCS_RNG.1							X	X	
FCS_COP.1/SHA								X	
FCS_COP.1/COS.3TDES								X	X
FCS_COP.1/COS.AES								X	X
FCS_COP.1/COS.RMAC								X	X
FCS_CKM.1/3TDES_SM							X	X	X
FCS_CKM.1/AES.SM							X	X	
FCS_CKM.1/RSA							X	X	
FCS_CKM.1/ELC							X	X	
FCS_COP.1/COS.CMAC								X	
FCS_COP.1/COS.RSA.S								X	
FCS_COP.1/COS.RSA.V								X	
FCS_COP.1/COS.ECDSA.S								X	
FCS_COP.1/COS.ECDSA.V								X	
FCS_COP.1/COS.RSA								X	
FCS_COP.1/COS.ELC								X	
FCS_CKM.4							X		
FCS_COP.1/PAUTH								X	
FTP_ITC.1/TC									X

Table 34 –Mapping between security objectives for the TOE and SFR

7.4.2 TOE Security Requirements Sufficiency

A detailed justification required for *suitability* of the security functional requirements to achieve the security objectives is given below.

The security objective **O.Integrity** “Integrity of internal data” requires the protection of the integrity of user data, TSF data and security services. This objective is addressed by the SFRs FDP_SDI.2/internal,, FDP_SDI.2/ReadEF , FPT_FLS.1 and FPT_TST.1: FPT_TST.1 requires self tests to demonstrate the correct operation of the TSF and its protection capabilities. FDP_SDI.2/internal and FDP_SDI.2/ReadEF require the TSF to monitor user data stored in containers and to take assigned action when data integrity error are detected. In case of failures, FPT_FLS.1 requires the preservation of a secure state in order to protect the user data, TSF data and security services.

The security objective **O.Confidentiality** “Confidentiality of internal data” requires the protection of the confidentiality of sensitive user data and TSF data. This objective is addressed by the SFRs FDP_RIP.1, FPT_FLS.1, FPT_EMS.1, FPT_TST.1 and FMT_MTD.1/NE: FMT_MTD.1/NE restricts the ability to export sensitive TSF data to dedicated roles, some sensitive user data like private authentication keys are not allowed to be exported at all. FPT_EMS.1 requires that the TOE does not emit any information of sensitive user data and TSF data by emissions and via circuit interfaces. Further, FDP_RIP.1 requires that residual information regarding sensitive data in previously used resources will not be available after its usage. FPT_TST.1 requires self tests to demonstrate the correct operation of the TSF and its confidentiality protection capabilities. In case of failures, FPT_FLS.1 requires the preservation of a secure state in order to protect the user data, TSF data and security services.

The security objective **O.Resp-COS** “Treatment of User and TSF Data” requires the correct treatment of the user data and TSF data as defined by the TSF data of the object system. This correct treatment is ensured by appropriate self tests of the TSF. FPT_TST.1 requires self tests to demonstrate the correct operation of the TSF and its data treatment.

The security objective **O.TSFDataExport** “Support of TSF data export” requires the correct export of TSF data of the object system excluding confidential TSF data. This objective is addressed by the SFRs FPT_TDC.1, FPT_ITE.1 and FPT_ITE.2: FPT_ITE.2 requires the export of dedicated TSF data but restricts the kind of TSF data that can be exported. Hence, confidential data shall not be exported. Also, the TSF is required to be able to export the fingerprint of TOE implementation by the SFR FPT_ITE.1. For Card Verifiable Certificates (CVC), the SFR FPT_TDC.1 requires the consistent interpretation when shared between the TSF and another trusted IT product.

The security objective **O.Authentication** “Authentication of external entities” requires the support of authentication of human users and external devices as well as the ability of the TSF to authenticate itself. This objective is addressed by the following SFRs:

- FIA_SOS.1 requires that the TSF enforces the length of the secret of the password objects.
- FIA_AFL.1/PIN requires that the TSF detects repeated unsuccessful authentication attempts and blocks the password authentication when the number of unsuccessful authentication attempts reaches a defined number.

- FIA_AFL.1/PUC requires that the TSF detects repeated unsuccessful authentication attempts for the password unblocking function and performs appropriate actions when the number of unsuccessful authentication attempts reaches a defined number.
- FIA_ATD.1 requires that the TSF maintains dedicated security attributes belonging to individual users.
- FIA_UAU.1 requires the processing of dedicated actions before a user is authenticated.
- Any other actions shall require user authentication.
- FIA_UAU.4 requires the prevention of reuse of authentication data.
- FIA_UAU.5 requires the TSF to support user authentication by providing dedicated commands. Multiple authentication mechanisms like password based and key based authentication are required.
- FIA_UAU.6 requires the TSF to support re-authentication of message senders using a secure messaging channel.
- FIA_UID.1 requires the processing of dedicated actions before a user is identified. Any other actions shall require user identification.
- FIA_API.1 requires that the TSF provides dedicated commands to prove the identity of the TSF itself.
- FMT_SMR.1 requires that the TSF maintains roles and associates users with roles.
- FIA_USB.1 requires that the TSF associates dedicated security attributes with subjects acting on behalf of that user. Also, the TSF shall enforce rules governing changes of these security attributes by the implementation of commands that perform these changes.
- FMT_MTD.1/PIN requires that the TSF restricts the ability to change password objects by the implementation of dedicated commands and management functions.
- FMT_MSA.1/PIN requires that the TSF enforces the access control policy to restrict the ability to change, enable and disable and optionally perform further operations of security attributes for password objects. For that purpose the SFR requires management functions to implement these operations.
- FMT_MTD.1/Auth requires that the TSF restricts the ability to import device authentication reference data by the implementation of dedicated commands and management functions.
- FMT_MSA.1/Auth requires that the TSF enforces the access control policy to restrict the ability to read security attributes for the device authentication reference data. For that purpose the SFR requires management functions to implement this operation.

The security objective **O.AccessControl** “Access Control for Objects” requires the enforcement of an access control policy to restricted objects and devices. Further, the management functionality for the access policy is required. This objective is addressed by the following SFRs:

- FMT_SMR.1 requires that the TSF maintains roles and associates users with roles.
- FIA_USB.1 requires that the TSF associates dedicated security attributes with subjects acting on behalf of that user. Also, the TSF shall enforce rules governing changes of these security attributes by the implementation of commands that perform these changes.
- FDP_ACC.1/MF_DF requires that the TSF enforces an access control policy to restrict operations on MF and folders objects as well as applications performed by subjects of the TOE.
- FDP_ACF.1/MF_DF requires that the TSF enforce an access control policy to restrict operations on MF and fol ders objects as well as applications based on a set of rules

defined in the SFR. Also, the TSF is required to deny access to the MF object in case of "Termination state" of the TOE life cycle.

- FDP_ACC.1/EF requires that the TSF enforces an access control policy to restrict operations on EF objects performed by subjects of the TOE.
- FDP_ACF.1/EF requires that the TSF enforce an access control policy to restrict operations on EF objects based on a set of rules defined in the SFR. Also, the TSF is required to deny access to EF objects in case of "Termination state" of the TOE life cycle.
- FDP_ACC.1/TEF requires that the TSF enforces an access control policy to restrict operations on transparent EF objects performed by subjects of the TOE.
- FDP_ACF.1/TEF requires that the TSF enforce an access control policy to restrict operations on transparent EF objects based on a set of rules defined in the SFR. Also, the TSF is required to deny access to transparent EF objects in case of "Termination state" of the TOE life cycle.
- FDP_ACC.1/SEF requires that the TSF enforces an access control policy to restrict operations on structured EF objects performed by subjects of the TOE.
- FDP_ACF.1/SEF requires that the TSF enforce an access control policy to restrict operations on structured EF objects based on a set of rules defined in the SFR. Also, the TSF is required to deny access to structured EF objects in case of "Termination state" of the TOE life cycle.
- FDP_ACC.1/KEY requires that the TSF enforces an access control policy to restrict operations on dedicated key objects performed by subjects of the TOE.
- FDP_ACF.1/KEY requires that the TSF enforce an access control policy to restrict operations on dedicated key objects based on a set of rules defined in the SFR. Also, the TSF is required to deny access to dedicated key objects in case of "Termination state" of the TOE life cycle.
- FDP_ACC.1/PKEYS requires that the TSF enforces an access control policy to restrict operations on dedicated perso key objects performed by subjects of the TOE.
- FDP_ACF.1/PKEYS requires that the TSF enforce an access control policy to restrict operations on dedicated perso key objects based on a set of rules defined in the SFR. Also, the TSF is required to deny access to dedicated perso key objects in case of "Termination state" of the TOE life cycle.
- FMT_MSA.3 requires that the TSF enforces an access control policy that provides re strictive default values for the used security attributes. Alternative default values for these security attributes shall only be allowed for dedicated authorized roles.
- FMT_SMF.1 requires tha t the TSF implements dedicated management functions that are given in the SFR.
- FMT_MSA.1/Life requires that the TSF enforces the access control policy to restrict the ability to manage life cycle relevant security attributes like lifeCycleStatus. For that purpose the SFRs require management functions to implement these operations.
- FMT_MSA.1/SEF requires that the TSF enforces the access control policy to restrict the ability to manage of security attributes of recorde. For that purpose the SFRs require management functions to implement these operations.
- FMT_MTD.1/PIN requires that the TSF restricts the ability to change password objects by the implementation of dedicated commands and management functions.
- FMT_MSA.1/PIN requires that the TSF enforces the access control policy to restrict the ability to read, change, enable, disable and optionally perform further operations of security attributes for password objects. For that purpose the SFR requires management functions to implement these operations.

- FMT_MTD.1/Auth requires that the TSF restricts the ability to import device authentication reference data by the implementation of dedicated commands and management functions.
- FMT_MSA.1/Auth requires that the TSF enforces the access control policy to restrict the ability to read security attributes for the device authentication reference data. For that purpose the SFR requires management functions to implement this operation.
- FMT_MTD.1/NE restricts the ability to export sensitive TSF data to dedicated roles, some sensitive user data like private authentication keys are not allowed to be exported at all.
- FMT_MTD.1/Ini and FMT_MTD.1/Pers, restrict the management of applications to authorised subjects and prevent the attempt to use management commands in order to bypass the access control policy.

The security objective **O.KeyManagement** "Generation and import of keys" requires the ability of the TSF to secure generation, import, distribution, access control and destruction of cryptographic keys. Also, the TSF is required to support the import and export of public keys. This objective is addressed by the following SFRs:

- FCS_RNG.1 requires that the TSF provides a random number generator of a specific class used for generation of keys.
- FCS_CKM.1/3TDES_SM, FCS_CKM.1/AES.SM, FCS_CKM.1/RSA,
- FCS_CKM.1/ELC, require that the TSF generates cryptographic keys with specific key generation algorithms as stated in the SFRs. The mentioned SFRs are needed to fulfil different requirements of the intended usage of the cryptographic keys.
- FCS_CKM.4 requires that the TSF destroys cryptographic keys in accordance with a given specific key destruction method.
- FDP_ACC.1/KEY and FDP_ACF.1/KEY controls access to the key management and the cryptographic operations using keys.
- FMT_MSA.1/Life requires restriction of the management of security attributes of the keys to subjects authorized for specific commands.
- FDP_ITC.1/PKEYS requires the perso keys loading for the personalization phase

The security objective **O.Crypto** "Cryptographic functions" requires the ability of the TSF to implement secure cryptographic algorithms. This objective is addressed by the following SFRs:

- FCS_RNG.1 requires that the TSF provides a random number generator of a specific class used for generation of keys.
- FCS_COP.1/SHA requires that the TSF provides different hashing algorithms that are referenced in the SFR.
- FCS_COP.1/COS.3TDES requires that the TSF provides decryption and encryption using 3TDES for secure messaging.
- FCS_COP.1/COS.AES requires that the TSF provides decryption and encryption using AES with different key sizes.
- FCS_COP.1/COS.RMAC requires that the TSF provides computation and verification of cryptographic checksums using the Retail MAC algorithm.
- FCS_COP.1/COS.CMAC requires that the TSF provides computation and verification of cryptographic checksums using the CMAC algorithm.
- FCS_COP.1/COS.RSA.S requires that the TSF provides the generation of digital signatures based on the RSA algorithm and different modulus' lengths.

- FCS_COP.1/COS.RSA.V requires that the TSF provides the verification of digital signatures based on the RSA algorithm and different modulus' lengths.
- FCS_COP.1/COS.ECDSA.S requires that the TSF provides the generation of digital signatures based on the ECDSA and different hash algorithms and different key sizes.
- FCS_COP.1/COS.ECDSA.V requires that the TSF provides the verification of digital signatures based on the ECDSA and different hash algorithms and different key sizes.
- FCS_COP.1/COS.RSA requires that the TSF provides encryption and decryption capabilities based on RSA algorithms with different modulus' lengths.
- FCS_COP.1/COS.ELC requires that the TSF provides encryption and decryption capabilities based on ELC algorithms with different key sizes.
- FCS_COP.1/PAUTH requires that the TSF provides encryption and decryption capabilities based on 2TDES and Retail MAC computation..
- FCS_CKM.1/3TDES_SM, FCS_CKM.1/AES.SM, FCS_CKM.1/RSA,
- FCS_CKM.1/ELC, require that the TSF generates cryptographic keys with specific key generation algorithms as stated in the SFRs. The mentioned SFRs are needed to fulfil different requirements of the intended usage of the cryptographic keys.

The security objective **O.SecureMessaging** "Secure messaging" requires the ability of the TSF to use and enforce the use of a trusted channel to successfully authenticated external entities that ensures the integrity and confidentiality of the transmitted data between the TSF and the external entity. This objective is addressed by the following SFRs:

- FCS_COP.1/COS.3TDES requires that the TSF provides decryption and encryption using 3TDES for secure messaging.
- FCS_COP.1/COS.AES requires that the TSF provides decryption and encryption using AES with different key sizes. One use case of that required functionality is secure messaging.
- FCS_COP.1/COS.RMAC requires that the TSF provides computation and verification of cryptographic checksums using the Retail MAC algorithm. One use case of that required functionality is secure messaging.
- FCS_CKM.1/3TDES_SM requires that the TSF generates cryptographic keys with specific key generation algorithms as stated in the SFR.
- FTP_ITC.1/TC requires that the TSF provides a communication channel between itself and another trusted IT product. The channel provides assured identification of its end points and protection of the channel data against modification and disclosure.

7.4.3 Dependency Rationale for Security Functional Requirements

The following table provides an overview how the dependencies of the security functional requirements are solved and a justification why some dependencies are not being satisfied.

SFR	dependent on	fulfilled by
FDP_RIP.1	No dependencies.	n. a.
FDP_SDI.2/ReadEF	No dependencies.	n. a.
FDP_SDI.2/Internal	No dependencies.	n. a.
FPT_FLS.1	No dependencies.	n. a.
FPT_EMS.1	No dependencies.	n. a.
FPT_TDC.1	No dependencies.	n. a.
FPT_ITE.1	No dependencies.	n. a.
FPT_ITE.2	No dependencies.	n. a.
FPT_TST.1	No dependencies.	n. a.
FIA_SOS.1	No dependencies	n.a.
FIA_AFL.1/PIN	FIA_UAU.1 Timing of authentication.	FIA_UAU.1
FIA_AFL.1/PUC	FIA_UAU.1 Timing of authentication.	FIA_UAU.1
FIA_ATD.1	No dependencies.	n. a.
FIA_UAU.1	FIA_UID.1 Timing of	FIA_UID.1
FIA_UAU.4	No dependencies.	n. a.
FIA_UAU.5	No dependencies.	n. a.
FIA_UAU.6	No dependencies.	n. a.
FIA_UID.1	No dependencies.	n. a.
FIA_API.1	No dependencies.	n. a.
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.1
FIA_USB.1	FIA_ATD.1 User attribute	FIA_ATD.1
FDP_ACC.1/MF_DF	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/ MF_DF
FDP_ACF. 1/ MF_DF	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/MF_DF, FMT_MSA.3
FDP_ACC.1/EF	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/EF
FDP_ACF.1/EF	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/EF, FMT_MSA.3
FDP_ACC.1/TEF	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/TEF

SFR	dependent on	fulfilled by
FDP_ACF.1/TEF	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/TEF, FMT_MSA.3
FDP_ACC.1/SEF	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/SEF
FDP_ACF.1/SEF	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/SEF, FMT_MSA.3
FDP_ACC.1/KEY	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/KEY
FDP_ACF.1/KEY	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/KEY, FMT_MSA.3
FDP_ACC.1/PKEYS	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1/PKEYS
FDP_ACF.1/PKEYS	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/PKEYS , FMT_MSA.3
FDP_ITC.1/PKEYS	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/PKEYS , FMT_MSA.3
FMT_MSA.3	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	FMT_MSA.1/Life, FMT_MSA.1/SEF, FMT_MSA.1/PIN, FMT_MSA.1/Auth, FMT_SMR.1
FMT_SMF.1	No dependencies.	n. a.
FMT_MSA.1/Life	[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	FDP_ACC.1/MF_DF, FDP_ACC.1/EF, FDP_ACC.1/TEF, FDP_ACC.1/SEF, FDP_ACC.1/KEY, FMT_SMR.1, FMT_SMF.1
FMT_MSA.1/SEF	[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	FDP_ACC.1/MF_DF, FDP_ACC.1/EF, FDP_ACC.1/TEF, FDP_ACC.1/SEF, FDP_ACC.1/KEY, FMT_SMR.1, FMT_SMF.1

SFR	dependent on	fulfilled by
FMT_MTD.1/PIN	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/Init	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/Perso	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FMT_MSA.1/PIN	[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	FDP_ACC.1/MF_DF, FDP_ACC.1/EF, FDP_ACC.1/TEF, FDP_ACC.1/SEF, FDP_ACC.1/KEY, FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/Auth	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FMT_MSA.1/Auth	[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	FDP_ACC.1/MF_DF, FDP_ACC.1/EF, FDP_ACC.1/TEF, FDP_ACC.1/SEF, FDP_ACC.1/KEY, FMT_SMR.1, FMT_SMF.1
FMT_MTD.1/NE	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FCS_RNG.1	No dependencies.	n. a.
FCS_COP.1/SHA	[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	The dependent SFRs are not applicable here because FCS_COP.1/SHA does not use any keys.

SFR	dependent on	fulfilled by
FCS_COP.1/COS.3TDES	[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_CKM.1/3TDES_SM, FCS_CKM.4
FCS_COP.1/COS.AES	[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_CKM.1/AES.SM, FCS_CKM.4
FCS_COP.1/COS.RMAC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction.	FCS_COP.1/COS.3TDES, FCS_CKM.4
FCS_CKM.1/3TDES_ SM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key desctruction	FCS_COP.1/COS.3TDES, FCS_CKM.4
FCS_CKM.1/AES.SM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction.	FCS_COP.1/COS.AES, FCS_CKM.4
FCS_CKM.1/RSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction.	FCS_COP.1/COS.RSA.S, FCS_COP.1/COS.RSA.V, FCS_COP.1/COS.RSA, FCS_CKM.4

SFR	dependent on	fulfilled by
FCS_CKM.1/ELC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction.	FCS_COP.1/COS.ELC, FCS_COP.1/COS.ECDSA.S, FCS_CKM.4
FCS_COP.1/COS.CMAC	[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_CKM.1/AES.SM, FCS_CKM.4
FCS_COP.1/COS.RSA.S	[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_CKM.1/RSA, FCS_CKM.4
FCS_COP.1/COS.RSA.V	[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_CKM.1/RSA, FCS_CKM.4
FCS_COP.1/COS.ECDSA.S	[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_CKM.1/ELC, FCS_CKM.4

SFR	dependent on	fulfilled by
FCS_COP.1/COS.ECDSA.V	[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	FMT_MTD.1/Auth requires import keys as of TSF data used by FCS_COP.1/COS.ECDSA.V (instead of import of user data FDP_ITC.1 or FDP_ITC.2) FCS_CKM.4
FCS_COP.1/COS.RSA	[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_CKM.1/RSA, FCS_CKM.4
FCS_COP.1/COS.EL C	[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_CKM.1/ELC, FCS_CKM.4
FCS_CKM.4	[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.1/3TDES_SM, FCS_CKM.1/AES.SM, FCS_CKM.1/RSA, FCS_CKM.1/ELC, ~~FCS_CKM.1/ DH.PACE~~
FCS_COP.1/PAUTH	[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	FDP_ITC.1/PKEYS FCS_CKM.4
FTP_ITC.1/TC	No dependencies.	n. a.
FRU_FLT.2/SICP	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1/SICP
FMT_LIM.1/SICP	FMT_LIM.2 Limited capabilities	FMT_LIM.2/SICP

SFR	dependent on	fulfilled by
FMT_LIM.2/SICP	FMT_LIM.1 Limited capabilities	FMT_LIM.1/SICP
FAU_SAS.1/SICP	No dependencies.	n. a.
FPT_PHP.3/SICP	No dependencies.	n. a.
FPT_ITT.1/SICP	FDP_ACC.1 1 Subset access control or FDP_IFC.1 Subset information flow control.	FDP_IFC.1/SICP
FDP_IFC.1/SICP	FDP_IFF.1 Simple security attributes	See [19]
FPT_ITT.1/SICP	No dependencies.	n. a.

Table 35 –Dependencies of the SFR

[19] Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the Data Processing Policy referred to in FDP_IFC.1 there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its Data Processing Policy (FDP_IFC.1)

7.4.4 Security Assurance Requirements Rationale

The current assurance package was chosen based on the pre-defined assurance package EAL4. This package permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level, at which it is likely to retrofit to an existing product line in an economically feasible way. EAL4 is applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security specific engineering costs.

Please refer section 6.3.3 “Rationale for the Assurance Requirements” in BSI-CC-PP-0035-2007 for the details regarding the chosen assurance level EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

The selection of the component ATE_DPT.2 provides a higher assurance than the pre-defined EAL4 package due to requiring the functional testing of SFR-enforcing modules. The functional

testing of SFR-enforcing modules is due to the TOE building a smartcard platform with very broad and powerful security functionality but without object system.

The selection of the component ALC_DVS.2 provides a higher assurance of the security of the development and manufacturing, especially for the secure handling of sensitive material. This augmentation was chosen due to the broad application of the TOE in security critical applications.

The selection of the component AVA_VAN.5 provides a higher assurance than the pre-defined EAL4 package, namely requiring a vulnerability analysis to assess the resistance to penetration attacks performed by an attacker possessing a high attack potential.
The set of assurance requirements being part of EAL4 fulfils all dependencies a priori.

The augmentation of EAL4 chosen comprises the following assurance components:
ATE_DPT.2,
ALC_DVS.2, and
AVA_VAN.5.

For these additional assurance component, all dependencies are met or exceeded in the EAL4 assurance package:

Component	Dependencies required by CC	Dependency fulfilled by
TOE security assurance requirements (only additional to EAL4)		
ALC_DVS.2	no dependencies	-
ATE_DPT.2	ADV_ARC.1	ADV_ARC.1
	ADV_TDS.3	ADV_TDS.3
	ATE_FUN.1	ATE_FUN.1
AVA_VAN.5	ADV_ARC.1	ADV_ARC.1
	ADV_FSP.4	ADV_FSP.4
	ADV_TDS.3	ADV_TDS.3
	ADV_IMP.1	ADV_IMP.1
	AGD_OPE.1	AGD_OPE.1
	AGD_PRE.1	AGD_PRE.1
	ATE_DPT.1	ATE_DPT.2

Table 36 –SAR Dependencies

8. TOE SUMMARY SPECIFICATION

The following sections describe the general technical mechanisms implemented by the TOE to meet all the requirements of the SFRs. Those are denoted in parentheses at the paragraphs that are related to them and again are listed in the last section with references to where they appear in the description, as a kind of index for the whole chapter.

8.1 Card Life Cycle State Machine

The ES incorporates a state machine to reflect the TOE life cycle phases. It ensures the secure evolution of the TOE from the IC manufacturing phase to the usage phase. Technically the life cycle state is an integrity-protected value stored in NVM, coding the life cycle states VIRGIN, MODULE, PERSO, and APPLICATIVE as specified in [GeGKOS_PERS]. The life cycle state machine operating on this state value has following properties:

(i) With the IC manufacturing process this life cycle state is unconditionally set to VIRGIN.

(ii) The life cycle evolves linearly in the sequence VIRGIN → MODULE → PERSO → APPLICATIVE (FMT_SMF.1) by successful execution of the production commands (see next section 8.2). The only way backwards is a switch from PERSO to MODULE by completely deleting the EEPROM content loaded so far (especially PIN and key values already personalized).

(iii) The main distinction in life cycles is the one between the productive phases (VIRGIN, MODULE, and PERSO) and the APPLICATIVE phase. Before APPLICATIVE phase only the production commands are available (FMT_MTD.1.1/Init, FMT_MTD.1.1/Perso). The switch to APPLICATIVE phase is irreversible; after this transition the applicative APDU commands are executable, but no longer the production commands. Technically the separation between production and application commands is accomplished by two different APDU dispatch routines.

8.2 Production Commands

The production of the TOE is accomplished via a dedicated set of production commands. Together with the Life Cycle State Machine they tie up the production flow as specified in [GeGKOS_PERS].

Each production command is implemented with a hard coded check for the necessary authentication state and the exact production phase(s) where it can be executed. Successful execution will process the life cycle state in a determined way.

(i) The VIRGIN state is given when starting the downloaded (flashed) ES for the first time. By writing protocol data the ES is advanced to MODULE state.

(ii) The loading of initialization data (Product Image) can only be executed in MODULE state and only after authentication with a dedicated, chip individual key only known to the TOE manufacturer (FMT_MTD.1.1/Init, FMT_SMF.1.1). After that authentication the pre-initialization and initialization flow is as follows:

- Loading a key for Product Image verification. This key is encrypted and integrity protected with the chip individual authentication key.
- Optionally loading filter code (“patches”) using command LoadFilter.
- Loading the keys for the personalization phase with command LoadSecret (FDP_ITC.1/PKEYS).

- Blockwise loading of the product image using command WriteImage.
- Authenticating the loaded image by sending a MAC computed with the key for image verification via command VerifyImage. Only if this last step is executed successfully, proving that the initialization data are authentic and integer, the life cycle state is advanced to PERSO.

(iii) In the PERSO state the personalization service provider authenticates himself using the personalization keys that were loaded by the TOE manufacturer in the initialization phase. The perso data are transmitted using the command StoreData (FMT_MTD.1.1/Perso, FMT_SMF.1.1). The image contains information which personalization data have to be loaded and which of them have to be sent encrypted. Upon completion of personalization the life cycle state is irreversibly switched to APPLICATIVE state and all the productive keys are deleted.

(iv) The personalization process follows the [EMV-CPS] scheme.

8.3 TOE IDENTIFICATION

(i) The command FINGERPRINT is implemented (FPT_ITE.1) in the ES according to Gematik specification, taking the option of computing SHA-256(prefix | M). The prefix is supplied with the command data, and M is a representative of all parts of OS and filter code.

(ii) In all life cycle states the card traceability data, stored during the production steps, can be retrieved via the GET DATA command. This includes chip serial number as well as platform and image identifiers.

(iii) The external Wrapper, delivered as java archive, provides its version number in the MANIFEST.MF file contained in its *.jar file.

8.4 OBJECT SYSTEM MANAGEMENT

(i) The TOE supports a hierarchical file system according to ISO 7816 with formatted and transparent EFs. In addition to files each folder can contain PIN, key, and rule objects.

(ii) Each file and object is equipped with a life cycle state and a set of references to rule objects, mediating the access to that resource. Formatted EFs may be configured to support record life cycle states. The management of those life cycle states is mediated by the access conditions of the relevant ISO commands (FMT_MSA.1/Life, FMT_MSA.1/SEF).

(iii) The following ISO commands are available file content access: APPEND RECORD, ERASE BINARY, ERASE RECORD, READ BINARY, READ RECORD, SEARCH RECORD, SET LOGICAL EOF, UPDATE BINARY, UPDATE RECORD, WRITE BINARY.

(iv) The following ISO commands are available for object management: SELECT, LOAD APPLICATION, DELETE, ACTIVATE, DEACTIVATE, TERMINATE CARD USAGE, TERMINATE DF, TERMINATE, ACTIVATE RECORD, DEACTIVATE RECORD, DELETE RECORD.

(v) There is always one currently selected folder and possibly a currently selected EF. Security environment numbers ranging from 1 to 4 can be active for each DF inside the currently selected path.

(vi) Only EFs of the currently selected folder can be accessed. Keys and PINs can only be referenced inside the currently selected path, therefore the hierarchical structure provides a means to separate applications and their data from each other.

(vii) The actual configuration of the object system will make up the card product, but is out of TOE scope. All the files and objects would be configured when being created by the initialization process or - in usage phase - by execution of the LOAD APPLICATION command.

(viii) The manufacturer proprietary command GET_OBJ_INFO is able to read out public (FMT_MTD.1/NE) configuration data of all objects other than files (FPT_ITE.2). It is used together with the external wrapper to export the data required for the official verification tool. Note that for files (folders and EFs) those data are exported via SELCT command, returning FCP data.

8.5 Random Numbers

For the cryptographic computations and authentication protocols described in the following sections the TOE has to generate random numbers that meet a defined quality metric. For this purpose the physical random generator of the hardware is used according to the chip manufacturers guidelines (FCS_RNG.1).

8.6 Cryptographic Computations

The ES contains a cryptographic library to implement the cryptographic procedures made available via the respective APDU commands. The basic operations for DES, AES, RSA and ELC are performed by the respective hardware co-processor. The following functionalities are implemented for this TOE:

(i) Hash computations (FCS_COP.1/SHA):

- SHA-256, SHA-384, SHA-512.
- SHA-1, only used in derivation of negotiated AES keys,

(ii) Key generation:

- Derivation of 3TDES keys in authentication protocols (FCS_CKM.1/3TDES_SM).
- Derivation of AES keys in authentication protocols (FCS_CKM.1/AES_SM).
- Onboard generation of RSA and ELC keys (FCS_CKM.1/RSA, FCS_CKM.1/ELC) with the command GENERATE ASYMMETRIC KEY PAIR.

(iii) Signature creation based on RSA with a preset key length of 2048 and 3072 bit (FCS_COP.1/COS.RSA.S):

- “ISO9796-2” scheme in the two modes DS1 and DS2,
- “PKCS#1v1_5” scheme, and
- “PKCS#1-PSS” scheme,

where exclusively the SHA-256 algorithm is used in internal hash computations (FCS_COP.1/HASH).

Signature creation based on ELC with all COS standard curves and corresponding hash algorithms SHA-256, SHA-384, and SHA-512 (FCS_COP.1/COS.ECDSA.S).

(iv) Import of public keys via PSO VERIFY CERTIFICATE transmitting the CVCs specified for the Health IT infrastructure, utilizing signature verification based on

- RSA in ISO9796-2 DS1 mode with a preset key length of 2048 bit (FCS_COP.1/COS.RSA.V) used with SHA-256 and

- ELC with the curves brainpoolP256r1, brainpoolP384r1, and brainpoolP512r1 used with corresponding hash algorithms SHA-256, SHA-384, and SHA-512 (FCS_COP.1/COS. ECDSA.V).

(v) Data deciphering with PKCS#1v1_5 padding and RSA OAEP for RSA keys (FCS_COP.1/COS.RSA) and with ELC keys (FCS_COP.1/COS.ELC).

(vi) Three-Key-TripleDES with a key length of 168 bit (3TDES) in following modes:
- 3TDES in CBC mode for message encryption for secure messaging (FCS_COP.1/COS.3TDES)
- RetailMAC for the symmetric authentication protocol, and for secure messaging (FCS_COP.1/COS.RMAC).

(vii) AES with a key length of 128 bit, 192 bit, and 256 bit in following modes:
- en-/decryption in CBC mode infor authentication protocols and for secure messaging (FCS_COP.1/COS.AES)
- CMAC for the symmetric authentication protocol, and for secure messaging (FCS_COP.1/COS.CMAC).

(viii) During production phases in secure environment the Two-Key-TripleDES with a key length of 112 bit is used for en/decryption in CBC mode and RetailMAC computation (FCS_COP.1/PAUTH).

8.7 PIN Authentication and PIN Object Management

(i) A human user can authenticate himself by correctly presenting a PIN value via the ISO APDU commands VERIFY, ENABLE VERIFICATION REQUIREMENT, DISABLE VERIFICATION REQUIREMENT or CHANGE REFERENCE DATA (FIA_UAU.5.1). The command parameters contain an identifier that refers to a corresponding PIN object in the object system. PIN objects have global authentication context when being child of the MF, or DF specific context otherwise (FIA_USB.1.1).

(ii) All PIN objects have a retry counter that is decremented on each unsuccessful authentication attempt and restored to a preset maximum (1 to 15) on successful attempts (FIA_AFL.1.1/PIN) and a minimum length.

(iii) On correct PIN presentation a security state associated with the PIN object is established (FIA_USB.1.3) in the global or the DF-specific list of PIN authentication states, depending on the PIN object's authentication context. It represents the user's identity (FIA_ATD.1.1, FIA_UAU.5.2, FMT_SMR.1.1) and corresponding access rights referenced in access rule objects (FMT_SMR.1.2).

(iv) After successive wrong PIN presentation exceeding the retry counter the PIN object is blocked so that no more PIN authentication can be achieved, even by presenting the correct PIN value (FIA_AFL.1.2/PIN).

(v) For each PIN object there is an associated unblocking code with a minimum length of 8 digits. For each Pin object the unblocking code can be used a preset number of times (1 to 15) to unblock the associated PIN in case it got blocked (FIA_AFL.1.1/PUC). The unblocking code is applied via the APDU command RESET RETRY COUNTER (FIA_AFL.1.2/PIN). After the last usage, regardless whether unsuccessful or not, the unblocking code itself gets irreversibly blocked and can no more be used then (FIA_AFL.1.2/PUC).

(vi) For PIN management the commands CHANGE REFERENCE DATA and RESET RETRY COUNTER, EN/DISABLE VERIFICATION REQUIREMENT are available (variants with and without presenting the old secret (FMT_MTD.1/PIN)). The current PIN status can be retrieved via command GET PIN STATUS (FMT_MSA.1.1/PIN).

(vii) If a user sets a new PIN secret via CHANGE REFERENCE DATA or RESET RETRY COUNTER, only values satisfying the predefined minimum and maximum lengths are accepted (FIA_SOS.1).

(viii) Directly after card production a PIN might be in transport state, depending on the personalization data. In this state it is not possible to establish the security state for that PIN. The card holder first has to replace the transport PIN by his preferred PIN, which must have at least the minimum PIN length preset. Only after this replacement the security state for this PIN can be set. It is not possible to switch the PIN back to transport state. The ISO command CHANGE REFERENCE DATA is used to replace the PIN. It is also used to select a new PIN value by presenting the old value, what restricts that operation to the card holder (FMT_MTD.1.1/PIN).

8.8 Asymmetric DeviceAuthentication

Asymmetric authentication is used by external devices to prove their authenticity to the card and optionally to secure the subsequent communication. The authentication protocol is as follows:

(i) If the public key of the external component's CA is not available inside the TOE, the corresponding certificate (containing that key) is entered (via APDU command PSO VERIFY CERTIFICATE). On successful certificate check according to functional specification (FPT_TDC.1) with the root key the public key of the external component's CA is stored in the TOE.

(ii) The certificate of the external component's public key is entered. On successful certificate check according to functional specification (FPT_TDC.1) with the CA key the public key of the external component is stored in the TOE, together with the CHA (in RSA case) or CHAT (in ELC case) from the CVC (FIA_USB.1.1). By the name of the entered key (CHR from the CVC) the external component is identified (FIA_UAU.1.1, FMT_SMR.1.1) and that key is selected for use in the subsequent authentication protocol via a MANAGE SECURITY ENVIRONMENT command.

(iii) If an authentication state already exists for the selected key it is deleted (FIA_USB.1.3).

(iv) With the command sequence INTERNAL AUTHENTICATE, GET CHALLENGE, EXTERNAL AUTHENTICATE in RSA case (FIA_UAU.5.1, FIA_API.1) using a mutual asymmetric, one time challenge-response authentication is performed (FIA_UAU.4.1).
In ELC case the authentication protocol is done with a two times execution of GENERAL AUTHENTICATE (FIA_UAU.5.1, FIA_API.1) in APDU chaining mode involving a strictly ascending scenario number (FIA_UAU.4.1).

A successful authentication has following effects:

(v) The authentication state for the entered external public key is set, representing the corresponding access rights: CHA from an RSA CVC and CHAT from an ECC CVC with effective access rights from the CVC chain (FIA_USB.1.3, FIA_ATD.1.1, FIA_UAU.5.2, FMT_SMR.1.2). That authentication state is evaluated when checking the external access to TSF data.

(vi) If indicated by the algorithm selected for the authentication protocol, volatile session keys are negotiated from the random numbers exchanged (FCS_CKM.1.1/SM) to secure the subsequent communication via Secure Messaging (FIA_UAU.5.1).

8.9 Symmetric Device Authentication

External devices can also authenticate themselves by a symmetric one-time challenge-response protocol with the command sequence GET CHALLENGE and MUTUAL AUTHENTICATE (FIA_UAU.5.1, FIA_API.1).

(i) Before executing that protocol the external device identifies itself by selecting the corresponding symmetric key object via a MSE-Set command with appropriate key identifier (FIA_UAU.1.1, FMT_SMR.1.1). Symmetric key objects have global authentication context when being child of the MF, or DF specific context otherwise (FIA_USB.1.1). If an authentication state already exists for that key it is deleted (FIA_USB.1.3).

(ii) With the selected key the symmetric authentication protocol is performed (utilizing FCS_RNG.1, FCS_COP.1/COS.AES or _COP.1/COS.3TDES). The involved challenge prevents the reuse of a successful authentication attempt (FIA_UAU.4.1).

A successful authentication has following effects:

(iii) An authentication state linked to the selected key object is set in the global or the DF-specific list of key authentication states, depending on the key object's location in the object system. (FIA_USB.1.1, FIA_USB.1.3) It identifies and represents the corresponding device with its access rights (FIA_ATD.1.1, FIA_UAU.5.2, FMT_SMR.1.2). That authentication state is evaluated when checking the external access to the TSF data.

(iv) Volatile session keys are negotiated from the random numbers exchanged (FCS_CKM.1/3TDES_SM, FCS_CKM.1/AES.SM) to secure the subsequent communication via Secure Messaging (FIA_UAU.5.1).

8.10 Access Management in Productive Phases

In productive phases the access check is hard wired within the production commands (FMT_MTD.1, FMT_SMR.1.1, FDP_ACC.1/PKEYS, FDP_ACF.1/PKEYS) and determined by the life cycle state, see sections 8.1 and 8.2. An authentication with K_{ICC} is necessary to be able to perform the card initialization. For personalization the authentication and subsequent secure messaging (conforming to CPS) has to be performed with the "Perso Keys" set by the Card Manufacturer. The Product Image is checked after loading by the card itself. The Developer is in charge for creating the Product Image MAC using the K_Verify.

8.11 Access Management in Usage Phase

As this product is a smart card complying with ISO 7816 the external world can only communicate with it via APDU commands. No direct access to the resources of the smart card, which in essence are file contents, PINs, and keys, is possible.

(i) In the usage phase every resource in the object system, i.e. each folder, EF, PIN object, key object, and rule object, is linked to a set of access rules to mediate the access to them (all iterations of FDP_ACC.1 and FMT_MSA.1). On each APDU command that affects a specific object, the relevant access rule is chosen from this set depending on

- the actual interface used (only contact interface is available for this TOE),

- the currently selected SE, and
- the object's life cycle state.

(ii) This access rule is evaluated before the intended functionality is invoked (all iterations of FDP_ACF.1.1, FMT_MTD.1/Auth). If no access rule is mapped for given SE and LCS, no access is granted in general (FDP_ACF.1.3). Following commands are an exception to this and in that case can be executed without restriction: SELECT, GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, LIST_PUBLIC KEY, GET_OBJ_INFO.

- The access rules of the usage phase consist of a Boolean combination of single "access conditions" (FDP_ACF.1.2). Those access conditions can specify:
- unrestricted access (ALWAYS) (FIA_UAU.1, FIA_UID.1),
- no access (NEVER),
- the presence of a device authentication state (FMT_SMR.1.2) referring to a key object,
- the presence of a PIN authentication state (FMT_SMR.1.2) referring to a PIN object
- The presence of secure messaging with volatile session keys established with a preceding device authentication (FIA_API.1.1),
- and any Boolean combination of those.

(iii) The TOE manages following lists of authentication states (FIA_USB.1.1):

- One global list for PIN authentication states (human users) in global context, each element referring to a successfully authenticated PIN object (and with it to its identifier),
- One DF-specific list for PIN authentication states (human users) in local context, each element referring to a successfully authenticated PIN object (and with it to its identifier).
- One global list for device authentication by symmetric and RSA keys in global context, each element referring to a successfully authenticated symmetric key object (and with it to its identifier) or successfully authenticated public RSA key object (and with it to its associated CHA).
- One DF-specific list for device authentication by symmetric and public RSA keys in local context, each element referring to a successfully authenticated symmetric key object (and with it to its identifier) or successfully authenticated public RSA key object (and with it to its associated CHA).
- One overall list for device authentication by public ELC keys, each element referring to a successfully authenticated public ELC key object (and with it to its associated CHAT).

(iv) If an authentication state gets out of scope by changing the currently selected path or by resetting the SE, it is deleted (FIA_USB.1.3):

8.12 SECURE MESSAGING

This component provides the functionality to ensure protection of the data exchanged via APDUs by authenticity, integrity, and confidentiality using 3TDES or AES cryptography.

(i) The authenticity and integrity is ensured by adding a Message Authentication Code (MAC) to the data (FCS_COP.1/COS.CMAC, FCS_COP.1/COS.RMAC).

(ii) The confidentiality is achieved by encrypting the exchanged data (FCS_COP.1/COS.AES, FCS_COP.1/COS.3TDES).

(iii) The Secure Messaging uses the volatile session keys that were negotiated in a preceding symmetric or asymmetric authentication protocol executed by an external device. Using these session keys the command data are equipped with a MAC and can optionally be encrypted (FTP_ITC.1.1).

(iv) Once the session keys are established to form a secure messaging channel with the authenticated external IT product, any command APDU may be sent by the external IT product with Secure Messaging using those session keys (FTP_ITC.1.2).

(v) The need to use Secure Messaging is governed by the access conditions set for the resource to be accessed. MAC and/or encryption must be present in command or response APDUs if listed in the access conditions, but may still be present if not listed. Only if the command message was sent in Secure Messaging format, the response messages will be in Secure Messaging format, too (FTP_ITC.1.3, FIA_API.1).

(vi) As long as the MAC is correctly verified and the encrypted data, if given, can successfully be decrypted with the presently stored session keys, the access conditions requiring secure messaging are fulfilled. In case of an error at MAC verification or decryption the session keys and the authentication state of the negotiation key, which was used to establish the session keys, are deleted (FIA_UAU.6, FIA_USB.1.3).

8.13 TSF Protection

The ES is designed to protect the TOE against fraudulent attacks. Supported by the security features of the platform (related to the SFRs of the IC platform) the following general mechanisms are in place:

(i) The TOE can only be started by a reset. On each reset the TOE is set to a secure state before the normal operation of the TSF starts, even after an unexpected abortion of TSF execution or TOE halt in response to attack detection (FPT_FLS.1.1). This includes the deletion of any session keys and security states established by authentication from users or components (FIA_USB.1.2, FIA_USB.1.3), see 8.11(iii). If a NVM update operation was interrupted for TSF data or EFs in transaction protected mode, a roll-back or roll-forward operation is performed to set the NVM to a consistent state. After finishing the startup the MF is selected, no EF is selected and all SE numbers are restored to #1 (FMT_MSA.3).

(ii) If during TSF execution an unexpected error occurs, the secure state of the TSF will be preserved by halting their execution. Such a halt state can only be left by a reset (FPT_FLS.1.1), what will set the TOE to a secure state again (see above).

(iii) Before the execution of the first APDU after start-up, the integrity of filter code (if present) is verified (FPT_TST.1.1).

(iv) During execution of the TSF at specific points it is checked if a relevant filter code is existing and in such a case it is executed (FPT_TST.1.1).

(v) The ES utilizes the hardware platform's protection and self check features like clock jitter or environmental sensors (FPT_TST.1.1).. Detection of faults leads to a TOE halt (FPT_FLS.1).

(vi) When retrieving random bytes from the hardware platform’s TRNG (classified as AIS31 P2 high) the corresponding validity flags are evaluated according to the platform’s security guidelines (FPT_TST.1.1). A detected fault would result in a TOE halt.

(vii) Before critical operations the ES executes a routine to check hardware registers and undisturbed hardware operation (FPT_TST.1.1). Detected faults would result in a TOE halt (FPT_TST.1.3). Also some desynchronization by software via random delay loops is done regularly.

(viii) All data in non-volatile memory are equipped with a checksum to detect integrity faults. While this feature can be deactivated for applicative data files at file creation time (Loading of the object system at initialization step or creation of objects via LOAD APPLICATION in usage phase), it cannot be deactivated for sensitive objects. File contents with integrity errors would still be exported on a read attempt, with a warning as response code though (FDP_SDI.2/ReadEF). But when accessing sensitive user or TSF data like PIN and key values, life cycle states, access conditions, and filter code, the TOE execution would be halted (FDP_SDI.2/Internal, FPT_TST.1.1, FPT_TST.1.2).

(ix) Security relevant data temporarily stored in RAM are also secured by a checksum: security states, session keys, external public keys, and transient RAM copies of non-volatile keys. In the case of an integrity error the TOE execution would be halted, muting the card (FDP_SDI.2/Internal, FPT_TST.1.1, FPT_TST.1.2).

(x) Session keys, RAM copies of private or secret keys, and volatile PIN data are explicitly erased as soon as they are no longer needed (FCS_CKM.4.1).

(xi) Sensitive data, especially keys and PIN values, are stored in a protected form: the data are masked so even in case an attacker succeeds in retrieving a memory dump those data are not available in plain

(xii) Sensitive operations like ELC, AES, RSA and 3TDES computations or PIN verification are programmed in a way that processing timing, electromagnetic radiation, or power consumption of the chip cannot be used to discover any PIN or secret/private key value (FPT_EMS.1).

(xiii) All sensitive code flows are secured by redundant branch checks, secure variable values, and execution tracing to permanently protect the TOE against physical tampering (FPT_TST.1.1). Any unexpected situation would lead to execution halt, muting the card (FPT_TST.1.3).

(xiv) The hierarchical object system handling of the ES provides a natural way to separate the data structures between applications (domain separation): An application is represented by a dedicated application DF and its child objects. EF data are not accessible from outside the current application DF and PIN and key objects are not accessible from outside the current path. Furthermore, the file system is completely separated from TSF internal data like counter measure configuration.

(xv) In case that an object (PIN, key, rule, DF, or EF) is explicitly deleted, the associated memory area is cleared directly at deletion time, making the previous information content unavailable (FDP_RIP.1.1).

8.14 COVERAGE OF SFRS

Requirement	Covering Location in Summary Specification
FDP_RIP.1	8.13(xv)
FDP_SDI.2/ReadEF	8.13(viii)
FDP_SDI.2/Internal	8.13(viii)(ix)
FPT_FLS.1	8.13(i)(ii)(v)
FPT_EMS.1	8.13(xii)
FPT_TDC.1	8.8(i)(ii)
FPT_ITE.1	8.3(i)
FPT_ITE.2	8.4(viii)
FPT_TST.1	8.13(iii)
FIA_SOS.1	8.7(vii)
FIA_AFL.1/PIN	8.7(ii)(iv)(v)
FIA_AFL.1/PUC	8.7(v)
FIA_ATD.1	8.7(iii), 8.8(v), 8.9(iii)
FIA_UAU.1	8.8(ii), 8.9(i), 8.11(ii)
FIA_UAU.4	8.8(iv), 8.9(ii)
FIA_UAU.5	8.7(i)(iii), 8.8(iv)(v)(vi), 8.9(iii)(iv)
FIA_UAU.6	8.12(vi)
FIA_UID.1	8.11(ii)
FIA_API.1	8.8(iv), 8.9, 8.11(ii), 8.12(v)
FMT_SMR.1	8.7(iii), 8.8(ii)(v), 8.9(i)(iii), 8.100, 8.11(ii)
FIA_USB.1	8.7(i)(iii), 8.8(ii)(iii)(v), 8.9(i)(iii), 8.11(iii)(iv), 8.12(vi), 8.13(i)
FDP_ACC.1/MF_DF	8.11(i)
FDP_ACF.1/ MF_DF	8.11(ii)
FDP_ACC.1/EF	8.11(i)
FDP_ACF.1/EF	8.11(ii)
FDP_ACC.1/TEF	8.11(i)
FDP_ACF.1/TEF	8.11(ii)
FDP_ACC.1/SEF	8.11(i)
FDP_ACF.1/SEF	8.11(ii)
FDP_ACC.1/KEY	8.11(i)

FDP_ACF.1/KEY	8.11(ii)
FDP_ACC.1/PKEYS	8.100
FDP_ACF.1/PKEYS	8.100
FDP_ITC.1/PKEYS	8.2(ii)
FMT_MSA.3	8.13(i)
FMT_SMF.1	8.1(ii), 8.2(ii)(iii)
FMT_MSA.1/Life	8.4(ii)
FMT_MSA.1/SEF	8.4(ii)
FMT_MTD.1/PIN	8.7(vi)
FMT_MTD.1/Init	8.1(iii), 8.2(ii)
FMT_MTD.1/Perso	8.1(iii), 8.2(iii)
FMT_MSA.1/PIN	8.7(vi), 8.11(i)
FMT_MTD.1/Auth	8.11(ii)
FMT_MSA.1/Auth	8.11(i)
FMT_MTD.1/NE	8.4(viii)
FCS_RNG.1	8.5
FCS_COP.1/SHA	8.6(i)
FCS_COP.1/COS.3TDES	8.6(vi)
FCS_COP.1/COS.AES	8.6(vii)
FCS_COP.1/COS.RMAC	8.6(vi)
FCS_CKM.1/3TDES_ SM	8.6(ii)
FCS_CKM.1/AES.SM	8.6(ii)
FCS_CKM.1/RSA	8.6(ii)
FCS_CKM.1/ELC	8.6(ii)
FCS_COP.1/COS.CMAC	8.6(vii)
FCS_COP.1/COS.RSA.S	8.6(iii)
FCS_COP.1/COS.RSA.V	8.6(iv)
FCS_COP.1/COS.ECDSA.S	8.6(iii)
FCS_COP.1/COS.RSA	8.6(v)
FCS_COP.1/COS.ELC	8.6(v)
FCS_CKM.4	8.13(x)
FCS_COP.1/PAUTH	8.6(viii)
FTP_ITC.1/TC	8.12(iii)(iv)(v)

9. STATEMENT OF COMPATIBILITY BETWEEN COMPOSITE ST AND PLATFORM ST

9.1 SFR PART

The following table lists the SFRs that are declared in the M7892 B11 security target [ST IC], and separates them in relevant platform-SFRs (RP_SFR) and irrelevant platform-SFRs (IP_SFR).

The table also provides the link between the relevant platform-SFRs and the composite product SFRs.

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SFR	Composite product SFRs
FRU_FLT.2	**Limited fault tolerance**: The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1).	SF_PMA	X		FPT_FLS.1 FPT_PHP.3
FPT_FLS.1	**Failure with preservation of secure state**: The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur.	SF_PS, SF_PMA, SF_PLA, SF_CS	X		FPT_FLS.1 FPT_PHP.3
FMT_LIM.1	The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Limited capability and availability Policy.	SF_DPM	X		No direct link to any composite-product SFR - used "transparently"
FMT_LIM.2	The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Limited capability and availability Policy.	SF_DPM	X		No direct link to any composite-product SFR.-used "transparently"
FAU_SAS.1	The TSF shall provide the test process before TOE Delivery with the capability to store the Pre-Initialization Data and / or Initialization and / or supplements of the Security IC Embedded Software in the not changeable configuration page area and non-volatile memory.	SF_DPM	X		No direct link to any composite-product SFR.-used "transparently"
FPT_PHP.3	The TSF shall resist physical manipulation and physical probing, to the TSF by responding automatically such that the SFRs are always enforced.	SF_DPM, SF_PS, SF_PMA, SF_PLA, SF_CS	X		FPT_FLS.1 FPT_PHP.3
FDP_ITT.1	The TSF shall enforce the Data Processing Policy to prevent the disclosure of user data when it is transmitted between physically-separated parts of the TOE.	SF_DPM, SF_PS, SF_PMA, SF_PLA, SF_CS	X		FPT_FLS.1 FPT_PHP.3
FPT_ITT.1	The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.	SF_DPM, SF_PS, SF_PMA, SF_CS	X		FPT_FLS.1 FPT_PHP.3

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SFR	Composite product SFRs
	The different memories, the CPU and other functional units of the TOE (e.g.a cryptographic co-processor) are seen as separated parts of the TOE.				
FDP_IFC.1	The TSF shall enforce the Data Processing Policy on all confidential data when they are processed or transferred by the TSF or by the Security IC Embedded Software.	SF_PS, SF_PMA, SF_PLA	X		FPT_FLS.1 FPT_PHP.3
FCS_RNG.1	The TSF shall provide a physical random number generator that implements total failure test of the random source.	SF_CS	X		FCS_RNG.1
FPT_TST.2	The TSF shall run a suite of self tests *at the request of the authorized user* to demonstrate the correct operation of the *alarm lines and/or following environmental sensor mechanisms.*	SF_PMA, SF_CS	X		FPT_FLS.1 FPT_PHP.3
FDP_ACC.1	The TSF shall enforce the Memory Access Control Policy on all subjects (software running at the defined and assigned privilege levels), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy, i.e. privilege levels.	SF_DPM, SF_PMA, SF_PLA	X		FPT_FLS.1 FPT_PHP.3
FDP_ACF.1	The TSF shall enforce the Memory Access Control Policy to objects based on the following: Subject: - software running at the IFX, OS1 and OS2 privilege levels required to securely operate the chip. This includes also privilege levels running interrupt routines. - software running at the privilege levels containing the application software Object: - data including code stored in memories Attributes: - the memory area where the access is performed to and/or - the operation to be performed. The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: evaluate the corresponding permission control information of the relevant memory range before, during or after the access so that accesses to be denied can not be utilized by the subject attempting to perform the operation.	SF_DPM, SF_PMA, SF_PLA	X		No direct link to any composite-product SFR.-used “transparently”
FMT_MSA.1	The TSF shall enforce the Memory Access Control Policy to restrict the ability to change default, modify or delete the security attributes permission control information to the software running on the privilege levels.	SF_DPM, SF_PMA, SF_PLA	X		No direct link to any composite-product SFR.-used “transparently”
FMT_MSA.3	The TSF shall enforce the Memory Access Control Policy to provide well defined default	SF_DPM, SF_PMA, SF_PLA	X		No direct link to any composite-product

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SF R	Composite product SFRs
	values for security attributes that are used to enforce the SFP. The TSF shall allow any subject, provided that the Memory AccessControl Policy is enforced and the necessary access is therefore allowed, to specify alternative initial values to override the default values when an object or information is created.				SFR.-used "transparently"
FMT_SMF.1	The TSF shall be capable of performing the following security management functions: access the configuration registers of the MMU.	SF_DPM, SF_PMA, SF_PLA	X		No direct link to any composite-product SFR.-used "transparently"
FCS_COP.1/ DES	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Triple Data Encryption Standard (3DES) in the Electronic Codebook Mode (ECB), in the Cipher Block Chaining Mode (CBC), in the Blinding Feedback Mode (BLD) and in the Cipher Feedback Mode (CFB)and with cryptographic key sizes of 2 x 56 bit or 3 x 56 bit, that meet the following standards: National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES),NIST Special Publication 800-67, Version 1.1	SF_CS	X		FCS_COP.1 ES does not use these functionalities but the ES uses the hardware accelerators for cryptographic computations.
FCS_COP.1/ AES	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Advanced encryption Standard (AES) and cryptographic key sizes of 128 bit or 192 bit or 256 bit that meet the following standards: U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL),Advanced Encryption Standard (AES), FIPS PUB 197	SF_CS		X	FCS_COP.1 ES does not use these functionalities but the ES uses the hardware accelerators for cryptographic computations.
FCS_COP.1/ ECDSA	The TSF shall perform *signature generation and signature verification* in accordance with a specified cryptographic algorithm ECDSA and cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following *standard:* *Signature Generation: 1. According to section 7.3 in ANSI X9.62 - 2005 Not implemented is step d) and e) thereof. The output of step e) has to be provided as input to our function by the caller. Deviation of step c) and f): The jumps to step a) were substituted by a return of the function with an error code, the jumps are emulated by another call to our function. 2. According to sections 6.2 (6.2.2. + 6.2.3) in ISO/IEC 15946-2:2002 Not implemented is section 6.2.1: The output of 5.4.2 has to be provided by the caller as input to the function.*	FS_CS		X	ES does not use these functionalities.

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SFR	Composite product SFRs
	The jumps to step a) were substituted by a return of the function with an error code, the jumps are emulated by another call to our function. 2. According to sections 6.2 (6.2.2. + 6.2.3) in ISO/IEC 15946-2:2002 Not implemented is section 6.2.1: The output of 5.4.2 has to be provided by the caller as input to the function. *Signature Verification: 1. According to section 7.4.1 in ANSI X9.62–2005 Not implemented is step b) and c) thereof. The output of step c) has to be provided as input to our function by the caller. Deviation of step d):* *Beside noted calculation, our algorithm adds a random multiple of BasepointerOrder n to the calculated values u1 and u2. 2. According to sections 6.4 (6.4.1. + 6.4.3 + 6.4.4) in ISO/IEC 15946-2:2002 Not implemented is section 6.4.2: The output of 5.4.2 has to be provided by the caller as input to the function.*				
FCS_COP.1/ ECDH	The TSF shall perform *elliptic curve Diffie-Hellman key agreement* in accordance with a specified cryptographic algorithm *ECDH* and cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following *standard:* *1. According to section 5.4.1 in ANSI X9.63 - 2001 Unlike section 5.4.1.3 our, implementation not only returns the x-coordinate of the shared secret, but rather the x-coordinate and y-coordinate.* *2. According to sections 8.4.2.1, 8.4.2.2, 8.4.2.3, and 8.4.2.4 in ISO/IEC 15946-3:2002: The function enables the operations described in the four sections.*	FS_CS		X	ES does not use these functionalities.
FCS_COP.1/ SHA	The TSF shall perform hash-value calculation of user chosen data in accordance with a specified cryptographic algorithm SHA-2 and with cryptographic key sizes of none that meet the following standards: U.S. Department of Commerce / National Bureau of Standards Secure Hash Algorithm, FIPS PUB 180-3, 2008-October, section 6.2 SHA-256 and section 6.4 SHA-512.	FS_CS	X		FCS_COP.1 ES does not use these functionalities. But the ES uses the hardware accelerators for cryptographic computations.
FCS_COP.1/ RSA	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Rivest-Shamir-Adleman (RSA) and cryptographic key sizes 1024 - 4096 bits that meet the following Standards Encryption: According to section 5.1.1 RSAEP in PKCS v2.2 RFC3447,without 5.1.1.1.	FS_CS	X		FCS_COP.1 ES does not use these functionalities but the ES uses the hardware accelerators for cryptographic computations.

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SFR	Composite product SFRs
	Decryption (with or without CRT): According to section 5.1.2 RSADP in PKCS v2.2 RFC3447 for u = 2, i.e., without any (r_i, d_i, t_i), i >2, therefore without 5.1.2.2.b (ii)&(v), without 5.1.2.1. 5.1.2.2.a, only supported up to n < 22048 Signature Generation (with or without CRT):: According to section 5.2.1 RSASP1 in PKCS v2.2 RFC3447 for u = 2, i.e., without any (r_i, d_i, t_i), i >2, therefore without 5.2.1.2.b (ii)&(v), without 5.2.1.1. 5.2.1.2.a, only supported up to n < 22048 Signature Verification: According to section 5.2.2 RSAVP1 in PKCS v2.2 RFC3447, without 5.2.2.1.				
FCS_CKM.1/ RSA	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm rsagen1 (PKCS v2.2 RFC3447) and specified cryptographic key sizes of 1024 – 4096 bits that meet the following standard: According to section 3.2(2) in PKCS v2.2 RFC3447, for u=2, i.e., without any (r_i, d_i, t_i), i > 2. For p x q < 22048 additionally according to section 3.2(1).	FS_CS		X	The TOE does not use the manufacturer library.
FCS_CKM.1/ EC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *Elliptic Curve EC specified in ANSI X9.62-2005* and *ISO/IEC 15946-1:2002* and specified cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following *standard:* ECDSA *Key Generation: 1. According to the appendix A4.3 in ANSI X9.62-2005 the cofactor h is not supported. 2. According to section 6.1 (not 6.1.1) in ISO/IEC 15946-1:2002*	FS_CS		X	The TOE does not use the manufacturer library.
FDP_SDI.1	The TSF shall monitor user data stored in containers controlled by the TSF for *inconsistencies between stored data and corresponding EDC* on all objects, based on the following attributes: *EDC value for the RAM, ROM and Infineon® SOLID FLASH™.*	SF_PMA	X		FPT_FLS.1 FPT_PHP.3
FDP_SDI.2	The TSF shall monitor user data stored in containers controlled by the TSF for *data integrity and one- and/or more-bit-errors* on all objects, based on the following attributes: *corresponding EDC value for RAM, ROM and Infineon® SOLID FLASH™ and error correction ECC for the Infineon® SOLID FLASH™.* Upon detection of a data integrity error, the TSF shall correct 1 bit errors in the Infineon®	SF_PMA	X		FPT_FLS.1 FPT_PHP.3

Platform SFR	Platform SFR Content	Platform SFR additional Information	RP_ SFR	IP_ SFR	Composite product SFRs
	SOLID FLASH™ automatically and inform the user about more bit errors.				

Table 37 – Composition – SFR part

SF_DPM : Device Phase Management
Transparent
SF_PS : Protection against snooping
Transparent
SF_PMA : Protection against Modifications attacks
The ES calls the UMSLC test e.g. before RSA crypto operations.
SF_PLA : Protection against logical attacks
Transparent
SF_CS : Cryptographic Support
The ES uses the hardware accelerators for cryptographic computations

9.2 OBJECTIVES

In the first column, the following table lists all relevant objectives for the TOE of the Platform ST. Corresponding objectives for the Composite TOE are assigned in the second column. The last column provides the result of the analysis for contradiction.

Platform-Objective	Corresponding Composite Objective	Result
TOE		
O.Phys-Manipulation	O.Phys-Manipulation	O.Phys-Manipulation of the Composite TOE is supported by O.Phys-Manipulation of the HW No contradiction to Composite-ST.
O.Phys-Probing	O.Phys-Probing	O.Phys-Probing of the Composite TOE is supported by O.Phys-Probing of the HW No contradiction to Composite-ST.
O.Malfunction	O.Malfunction	O.Malfunction of the Composite TOE is supported by O.Malfunction of the HW No contradiction to Composite-ST.
O.RND	O.RND	O.RND of the Composite TOE is supported by O.RND of the HW No contradiction to Composite-ST.
O.Leak-Inherent	O.Leak-Inherent	O.Leak-Inherent of the Composite TOE is supported by O.Leak-Inherent of the HW No contradiction to Composite-ST.
O.Leak-Forced	O.Leak-Forced	O.Leak-Forced of the Composite TOE is supported by O.Leak-Forced of the HW No contradiction to Composite-ST.
O.Abuse-Func	O.Abuse-Func	O.Abuse-Func of the Composite TOE is supported by O.Abuse-Func of the HW No contradiction to Composite-ST.
O.Identification	O.Identification	O.Identification of the Composite TOE is supported by O.Identification of the HW No contradiction to Composite-ST.
O.Add-Functions	No correspondence	Platform provides the following specific security functionality to the Embedded Software: • Advanced Encryption Standard (AES)

Platform-Objective	Corresponding Composite Objective	Result
		- Triple Data Encryption Standard (3DES), - Rivest-Shamir-Adleman (RSA) - Elliptic Curve Cryptography (EC) - Secure Hash Algorithm (SHA-2) But the ES not used these functionalities. No contradiction to Composite-ST.
O.Mem-Access	No correspondence	Platform provides capability to define restricted access memory area But the ES does not use these functionalities. No contradiction to Composite-ST.

9.3 THREAT

In the first column, the following table lists all relevant threats of the Platform ST, those are all threats, that are traced to the relevant TOE security objectives. Corresponding threats are assigned in the second column. The last column provides the result of the analysis for contradiction.

Platform-Threat	Corresponding Composite Threats	Result
T.Phys-Manipulation	T.Phys-Manipulation	T.Phys-Manioulation of the composite TOE is identical to T.Phys-Manipulation of the platform No contradiction to Composite-ST.
T.Phys-Probing	T.Phys-Probing	T.Phys-Probing of the composite TOE is identical to T.Phys-Probing of the platform No contradiction to Composite-ST.
T.Malfunction	T.Malfunction	T.Malfunction of the Composite TOE is identical to T.Malfunction of the platform No contradiction to Composite-ST.
T.Leak-Inherent	T.Leak-Inherent	T.Leak-Inherent of the Composite TOE is identical to T.Leak-Inherent of the platform No contradiction to Composite-ST.
T.Leak-Forced	T.Leak-Forced	T.Leak-Forced of the Composite TOE is identical to T.Leak-Forced of the platform No contradiction to Composite-ST.
T.Mem-Access	T.Forge_Internal_Data T.Compromise_Internal_Data	T.Forge_Internal_Data, T.Compromise_Internal_Data address T_Mem-Acces of the platform No contradiction to Composite-ST.
T.Abuse-Func	T.Abuse-Func	T.Abuse-Func of the Composite TOE is identical to T.Abuse-Func of the platform No contradiction to Composite-ST.
T.RND	T.RND	T.RND of the Composite TOE is identical to T.RND of the platform No contradiction to Composite-ST.

Table 38 – Composition – Threats part

9.4 OSP Part

IC OSP label	IC OSP content	Link to the composite product
P.Process-TOE	Protection during TOE Development and Production:An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.	No contradiction with the present evaluation; the chip traceability information is used to identify the composite TOE.
P.Add-Functions	Additional Specific Security Components: The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: • Advanced Encryption Standard (AES) • Triple Data Encryption Standard (3DES) • Rivest-Shamir-Adleman Cryptography (RSA), • Elliptic Curve Cryptography (EC) • ▯ Secure Hash Algorithm SHA-2	The ES uses the hardware accelerators for cryptographic computations.

Table 39 – Composition – OSPs part

9.5 Assumptions Part

Please refer to chapter 4.4

IC assumption label	IC assumption title	IrPA	CfPA	SgPA	Link to the composite product
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation	X			A.Process-Sec-SC
A.Plat-Appl	Usage of Hardware Platform		**X**		Fulfilled by the composite-SAR ADV_COMP.1 (cf [CCDB], Appendix 1.2, §72 and §73)
A.Resp-Appl	Treatment of User Data		**X**		A.Resp-ObjS
A.Key-Function	Usage of key-dependent functions		**X**		O.KeyManagement

Table 40 – Composition – Assumptions part

IrPA means "*The assumptions being not relevant for the Composite-ST, e.g. the assumptions about the developing and manufacturing phases of the platform.*"
CfPA means "*The assumptions being fulfilled by the Composite-ST automatically. Such assumptions of the Platform-ST can always be assigned to the TOE security objectives of the Composite-ST. Due to this fact they will be fulfilled either by the Composite-TSF or by the Composite-TAM automatically.*"
SgPA means "*The remaining assumptions of the Platform-ST belonging neither to the group IrPA nor CfPA. Exactly this group makes up the significant assumptions for the Composite-ST, which shall be included into the Composite-ST.*"

9.6 SECURITY OBJECTIVES FOR THE ENVIRONMENT PART

See chapter 5.3

9.7 ASSURANCE REQUIREMENTS PART

The IC is EAL 6 augmented with ALC_DVS.2 and AVA_VAN.5.
The composite product is EAL4 augmented with:

- AVA_VAN.5 Advanced methodical vulnerability analysis
- ALC_DVS.2 Development security
- ATE_DPT.2 Test depth

There's no contradiction to composite ST.

10. ABBREVIATIONS

Name	Definition
AC	Access Conditions
ADF	Application DFs
ALR	Anomaly List Report
APC	Subsystem “APDU Container”
APDU	Application Protocol Data Unit
API	Application Programming Interface
APL	Acceptance Plan
ARGOS	Acceptance and Requirements for GEMALTO Organization System
ATM	Automatic Teller Machine
ATR	Answer To Reset
CAR	Card Acceptance Report
CC	Common Criteria (referenced as CC)
CEPS	Common Electronic Purse Specifications
CI	Configuration Item
CIS	Card Initialisation Specification
CLI	Command Line Interface
COS	Card Operating System
CM	Configuration Management
CMP	Configuration Mangement Plan
CMS	Configuration Management System
CSP	Certification-Service provider
CUD	Client User Document
CVC	Card Verifiable Certificate
DAR	DIL Acceptance Report
DF	Dedicated File
DIL	Dual In Line
EAL	Evaluation Assurance Level
EC	Electronic Cash
EEPROM	Electrically Erasable and Programmable Read Only Memory
EF	Elementary File
eGK	elektronische Gesundheitskarte
eHC	electronic Health Card
EMV	Europay-Mastercard-Visa
ERR	Subsystem “Error Handling”
ES	Embedded Software
FRS	Functional Requirement Specifications
FS	Subsystem “File System”
HAL	Subsystem “Hardware Abstraction Layer”
HBCI	HomeBanking Computer Interface
HEC	Health Employee Card (technically a type of HPC)
HSM	Hardware Security Module
HPC	Health Professional Card
IC	Integrated circuit

ID	Identifier
IFD	Interface device
INS	Instruction code
I/O	Input/Output
IT	Information Technology
IUD	Internal User Documentation
LRC	Longitudinal Redundancy Checksum
MAC	Message Authentication Code
MAR	Mask Acceptance Report
MF	Master File
OS	Operating System
OSP	Operational Security Policy
*OSP.****	Naming convention for organisational security policies in this ST, e. g. OSP.User_Information
*OT.****	Naming convention for security objectives for the TOE in this ST, e. g. OT.Access_Rights
PIN	Personal Identification Number (authentication feature)
PKI	Public Key Infrastructure
PL	Project Leader
PP	Protection Profile
PROC	Subsystem “Process Handling”
PUC	PIN Unblocking Code
PVCS	Product Version Control System
RAD	Reference Authentication Data
RAM	Random Access Memory
ROM	Read Only Memory
SAR	Security assurance requirements
RSA	Rivest Shamir Adleman (algorithm)
SCM	Software Configuration Mangement
SCMA	Software Configuration Mangement Administrator
SCU	Smart Card Utility
SDD	Software Design Description
SDD1	Preliminary Software Design Description
SDD2	Detailed Software Design Description
SDO	Signed Data Object
SF	Security Function
SFP	Security Function Policy
SFP_access_rules	Name of the security functional policy defining the access rights to assets (data) in the TOE. It is defined in OT.Access_Rights and used by access control SFRs
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SMS	Software Masking Specification
SOF	Strength Of Function
SK	Subsystem “Security Kernel”
SM	Secure Messaging
SMC	Security Module Card
ST	Security Target
SVA	Software Validation Approval

TBX	Subsystem “Toolbox”
TDM	Technical Data Management
TOE	Target of Evaluation
TOE_App	Application Part of the TOE
TOE_ES	TOE Embedded Software (operating system of the TOE)
TOE_IC	The integrated circuit of the TOE, the hardware part together with IC dedicated software
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
UART	Universal Asynchronous Receiver Transmitter
UTP	Unitary Test Plan
UTR	Unitary Test Report
VAD	Verification authentication data
VCC	Voltage at the Common Collector
VLR	Validation Review
VTP	Validation Test Plan
VTP1	Preliminary Validation Test Plan
VTP2	Detailled Validation Test Plan
VTR	Validation Test Report
VTS	Validation Test Specification
X.509	A certificate format

Table 41 – Abbreviation table

11. GLOSSARY

The glossary elements for this development project are given in the table below:

Administrator means an user authorized to the TOE for personalisation, or other TOE administrative functions.

Archive. PVCS or VSS file which contains the evolution history of a work file. PVCS or VSS is able to rebuild any revision of the work file. Historical information includes description of changes, who made them, and when they were made. The archive also contains information about the status and attributes of the archive and its associated work file

Authentication data is information used to verify the claimed identity of a user.

Branch. Separate line of development consisting of one or more revisions that diverge from a revision on the trunk or from another development branch

Check-In. Action of storing a new revision in an archive.

Check-Out. Action of getting a revision from an archive. Then the archive is locked, and can be modified to do another revision.

Component. The hardware component of the Operating System.

Evolution Index (VSS). Symbolic reference used to uniquely identify a preliminary software version.

Evolution Index (PVCS). This number (integer) is used to uniquely identify a software version. Take note that the EI is different from the revision number that is automatically generated by PVCS.

Filter. A set of bug fixes and adjustments of the ROM code, residing in EEPROM

Folder (VSS/PVCS). A folder enables to organise archives in the Version Manager MMI. It logically links some archives

IC dedicated software. The part of the TOE's software, which is provided by the hardware manufacturer

IC Dedicated Support Software. That part of the IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Software might be restricted to certain phases.

IC Dedicated Test Software. That part of the IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.

Initialisation Data. Any data defined by the TOE Manufacturer and injected into the non-volatile memory by the Integrated Circuits manufacturer (Phase 2). These data are for instance used for traceability and for IC identification (IC identification data).

Integrated circuit (IC) Electronic component(s) designed to perform processing and/or memory functions. The eHC's chip is a integrated circuit.

Label. Symbolic name assigned to a revision in one or more archives. Labels provide a convenient way to refer to several archives with different revisions by a single name

Mask. Software developed by GEMALTO to be implemented in the chip

Module. Subset of commands and/or mechanisms. A module groups several routines allowing a logical function. A module cannot be broken up. Most of the time, a module will contain only one source file in the OS referential while it may involve several tests in the Test referential. [examples of modules for the Administrative Kernel brick are Record, Authentication, Secure Messaging, ...]

Mutual Authentication. Type of those cryptographic protocols, were two entities mutually verify the authenticity of each other, for smart cards this is realised by suitable sequences of amt card commands and responses
Personalisation. The process by which personal data are brought into the TOE before it is handed to the card holder
Product. Set of modules that constitute a final mask or a final filter (final release)
Project. See VSS/PVCS project
Reference authentication data (RAD) means data persistently stored by the TOE for verification of the authenti164uthorizedempt as authorised user.
Referential. Set of software components which are used by several Teams such as the OS software or the Test environment. The Referential contains all the archives of a project
Revision. Particular iteration of a work file in an archive. Each time a work file is modified and checked back into the archive, VSS/PVCS creates a new revision and assigns it automatically a new revision number
Rule_*. Naming convention for access control rules in this ST, defined in SFP_access_rules.
Secure Channel. A connection between two devices, which is secured against interception or modification of the transmitted data. The TOE realises a secure channel to other devices using secure messaging.
secure messaging in encrypted mode. Secure messaging using encryption and message authentication code according to ISO/IEC 7816-4
Service_**.** Services provided by the TOE (e. g. Service_Privacy)
Signature attributes means additional information that is signed together with the user message.
Sub-Referential. Consistent set of software components (Example: test scripts, specification documents,). A Sub-referential belongs to a Referential.
Tip Revision. The latest revision of a line of development (the trunk or a branch)
TSF data. Data created by and for the TOE, that might affect the operation of the TOE
User means any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data. Data created by and for the user, that does not affect the operation of the TSF
Verification authentication data (VAD) means authentication data provided as input by knowledge or authentication data derived from user's biometric characteristics.
VSS/PVCS Project. Logical set of folders and archives
Work File. Copy of an archive revision, usually for working with it on a local PC. If the archive is "checked out" this copy can be modified and "checked in" again as the new revision of the archive.
Work File Directory. Local folder to hold the archive copies generated by "Check Out" or "Get" actions (in German: "Auscheckordner"). A folder in VSS must be linked to a work file directory, so that "Get" actions can be performed.

Table 42 – Glossary table

12. REFERENCES

The documents and reference elements for this development project are given in the table below:

Reference	Title of document	Author
Common Criteria Documents		
CCPART1	Common Criteria for Information Technology Security Evaluation. Part 1: Introduction & general model, CCMB-2012-09-001. Version 3.1. Rev4 September 2012.	Common Criteria Project Sponsoring Organizations
CCPART2	Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional requirements, CCMB-2012-09-002. Version 3.1. Rev 4 September 2012.	Common Criteria Project Sponsoring Organizations
CCPART3	Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance requirements, CCMB-2012-09-003. Version 3.1. Rev 4 September 2012.	Common Criteria Project Sponsoring Organizations
CEM	Common Methodology for Information Technology Security Evaluation CCMB-2012-09-004, version 3.1 Rev 4, September 2012.	Common Criteria Project Sponsoring Organizations
JIL	Application of attack potential to smartcards, version 2.9 January 2013	Joint Interpretation Library
AIS 34	Application Notes and Interpretation of the Scheme (AIS), AIS 34, Evaluation Methodology for CC Assurance Classes for EAL5+, Version 3., 03.09.2009, Bundesamt für Sicherheit in der Informationstechnik.	BSI
AIS 36	Composite product evaluation for Smart Cards and similar devices, Version 1, Rev 1, September 2007, CCDB-2007-09-001	Common Criteria
AAPSC	Application of Attack Potential to Smartcards, Version 2.7, February 2009	Common Criteria Project Sponsoring Organizations
AMSRP	Attack Methods for Smartcards and Similar Devices, Version 1.5, February 2009	Common Criteria Project Sponsoring Organizations
ETR_Lite Annex A	ETR-lite for composition: Annex A Composite smartcard evaluation : Recommended best practice, Version 1.2, March 2002	Common Criteria Project Sponsoring Organizations
PP eHC	Protection Profile BSI-CC-PP-0082-V2: Card Operating System Generation 2 (PP COS G2) Version 1.9 of 18th November 2014	BSI
GeGKOS_PERS	Perso Manual for GEGKOS C1,version 1.12 of 22/09/2016	Gemalto
EMV-CPS	EMV card personalization specification, Version 1.1,July 2007.	EMV
Chip Documents		

ST IC	Security Target Lite M7892 B11 Recertification Including optional software libraries RSA – EC – SHA 2 – Toolbox, Version 0.3 - 2015-10-13	Infineon technologies AG
CER IC	BSI-DSZ-CC-0782-V2-2015: Infineon Security Controller M7892 B11 with optional RSA2048/4096 v1.02.013, EC v1.02.013,SHA-2 v1.01 and Toolbox v1.02.013 libraries and with specific IC dedicated software (firmware) Certification date : 03/11/2015	BSI
DB IC	SLx 70 family Hardware Reference Manual, Nov 2010	Infineon
eHC Documents		
gemSpec_COS	Spezifikation des Card Operating System (COS), Elektrische Schnittstelle, Version 3.8.0 17.07.2015, which includes [gemErrata_R1.4.7] v. 24.07.2015, [gemErrata_R1.4.7_200] v. 30.09.2015 [gemErrata_R1.4.5] v. 07.05.2015	gematik
GemSpec_COS-wrapper	Spezifikation Wrapper, Version 1.7.0, 17.07.2015	gematik
Reference	**Title of document**	**Author**
ISO Documents		
ISO 7810	ISO 7810: Identification cards - Physical characteristics. 2003	ISO
ISO C1	ISO 7816 – 3, Identification cards - Integrated circuit(s) cards with contacts, Part 1: Physical characteristics. 2006	ISO
ISO C3	ISO 7816 - 3, Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols 2006	ISO
ISO C4	ISO 7816 - 4, Identification cards - Integrated circuit(s) cards with contacts, Part 4: Inter-industry commands for interchange 2013	ISO
ISO C4'	ISO 7816 - 4, Identification cards - Integrated circuit(s) cards with contacts, Part 4: Inter-industry commands for interchange, AMENDMENT 1: Impact of secure messaging on the structures of APDU messages. 2013	ISO
ISO C8	ISO 7816 - 8, Identification cards - Integrated circuit(s) cards with contacts, Part 8: Security related inter-industry commands. 2004	ISO
ISO C9	ISO 7816 - 9, Identification cards - Integrated circuit(s) cards with contacts 2004	ISO
9796-2	ISO/IEC 9796-2:2010 Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms, 2010-12, ISO	ISO
ISO HF3	ISO 10118 - 3, Information technology - Security techniques - Hash-functions, Part 3: Dedicated hash functions, 1998	ISO
RSA Laboratories Documents		

PKCS1	PKCS#1 RSA Cryptography Standard . Version 2.2 October 27, 2012	RSA Laboratories
Nist Document		
FIPS 46-3	Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3) of U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology Data encryption standard (DES and TDES) – Reaffirmed 1999 October 25	NIST
FIPS 197	FIPS 197: AES Advanced Encryption Standard. National Institute of Standards and Technology – Nov 2001	NIST
Hash document		
FIPS 180-3	FIPS 180-3 Secure Hash Standard (SHS) - October 2008	NIST
FIPS 180-4	FIPS 180-4: Secure Hash Standard (SHS) – March 2012	NIST
Random generators		
AIS31	Functionality classes for random number generators, Version 2.0, September 18,2011. http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf	BSI
Technical Guidelines		
TR-03111	Technical Guideline TR-03111 Elliptic Curve Cryptography, TR-03111, version 2.0, 28.08.2012,	BSI
TR-03116	Technische Richtlinie TR-03116, eCard-Projekte der Bundesregierung, Version 3.19 vom 03.12.2015	BSI
TR-03143	Technische Richtlinie TR-03143 „eHealth G2-COS Konsistenz-Prüftool“ Version 1.0 vom 08.05.2015	BSI

Table 43 – Reference table

<END OF DOCUMENT>