T · · Systems ·

Specification of the Security Target TCOS FlexCert Version 2.0 Release 1/SLE78CLX1440P

Version: 2.0.1/20150605

Dokumentenkennung:	CD.TCOS.ASE
Dateiname:	ASE TCOS FlexCert Version 2.0 Release 1 (IFX).docx
Stand:	05.06.2015
Version:	2.0.1
Hardware Basis:	SLE78CLX1440P
Autor:	Ernst-G. Giessmann
Geltungsbereich:	TeleSec Entwicklungsgruppe
Vertraulichkeitsstufe:	**Öffentlich**

History

Version	Date	Remark
2.0.1	2015-06-05	Final Document

Contents

1 ST Introduction

1 This section provides document management and overview information that are required a potential user of the TOE to determine, whether the TOE fulfils her requirements.

1.1 ST Reference

2

Title:	Specification of the Security Target TCOS FlexCert Version 2.0 Release 1/SLE78CLX1440P
TOE:	TCOS FlexCert Version 2.0 Release 1/SLE78CLX1440P
Sponsor:	T-Systems International GmbH
Editor(s):	Ernst-G. Giessmann, T-Systems International GmbH, TeleSec
CC Version:	3.1 (Revision 4)
Assurance Level:	EAL4 augmented.
General Status:	Final Document
Version Number:	2.0.1
Date:	2015-06-05
Certification ID:	BSI-DSZ-CC-0904
Keywords:	Gesundheitskarte, electronic health card, TCOS

1.2 TOE Reference

3 This Security Target refers to the Product “TCOS FlexCert Version 2.0 Release 1” (TOE) of T-Systems International GmbH for CC evaluation.

1.3 TOE Overview

4 The Target of Evaluation (TOE) addressed by this Security Target is a smart card with contact based and contact-less interfaces implementing an Operating System without any object system. The TOE’s type is “Card Operating System Platform”.

5 The Operating System is based on the Specification of the Gesundheitskarte [EGK-COS]. Despite the fact, that the object system is not included in the TOE, it will nevertheless always be used with a specified object system. Depending on the object system initialization the smart card product will represent a ready for Personalization electronic Healthcare Card, Health Professional Card or a Secure Module Card of a specified type.

6 The TOE provides the following main security functionalities according to [EGK-COS]:

- authentication of human user and external devices;
- storage of and access control on user data;
- key management and cryptographic functions;

- management of TSF data including life cycle support;
- export of non-confidential TSF data of the object system if implemented.

7 The TOE is a ready for implementation of the object system consisting of the Master File (MF), the Dedicated Files (DF), Elementary Files (EF) and internal security objects including TSF data conforming to the ISO7816 standards.

8 The hardware bases on a Infineon chip SLE78CLX1440P with the TCOS operating system.

9 The cryptographic algorithms used by the TOE are defined outside the TOE. The security parameters of these algorithms must be selected by card issuer according to Security Policies [TR3116-1]. The TOE supports standardized domain elliptic curve parameters mentioned in [RFC5639] (key lengths 256, 384 and 512 bit) and the NIST P-256 and P-384 curves (key length 256 and 384 bit) mentioned in [FIPS186] including the corresponding hash functions. Integrity and Confidentiality of the communication is protected by symmetric cryptographic algorithms. The TOE provides AES and TDES[1] with corresponding key lengths of 128, 192, 256 and 168 bits.

10 The TOE's chip is integrated into a plastic, optically readable part of the Health Card. This is not part of the TOE.

11 In some context the hardware may be relevant, and if so, the TOE will be identified in more detail as "TCOS FlexCert Version 2.0 Release 1/SLE78CLX1440P", otherwise the notion "TCOS FlexCert Version 2.0 Release 1" will be used, indicating that this context applies to any realization regardless which hardware base is used. The chip SLE78CLX1440P is selected from the M7820 family. Note that the Chip Identifier Byte is not used in the TOE identification because it has no impact on the evaluation.

12 The TOE follows the composite evaluation aspects ([AIS36]). The Security Target of the underlying platform ([HWST]) claims conformance to Smartcard IC Platform Protection Profile ([PP0035]).

13 This composite ST is based on the ST of the underlying platform ([HWST]). The life cycle compatibility of the Life Cycle Model of the Protection Profile [PPCOS] and the Life Cycle Model required by [PP0035] will be shown in chapter 1.4.4.

1.4 TOE Description

1.4.1 TOE Definition

14 The TOE comprises of

- the circuitry of the chip including all IC Dedicated Software being active in the Operational Phase of the TOE (the integrated circuit, IC),
- the IC Embedded Software (Card Operating System, COS) including configuration and initialization data related to the security functionality of the chip,
- the associated guidance documentation including description of the file system installation procedure,
- a wrapper for interpretation of exported TSF data.

1 TDES is the notation for Triple DES according to [SP800-67], the Specification [EGK-COS] uses 3TDES instead.

15 The components of the TOE are therefore the hardware (IC) and the operating system TCOS (OS) ready for initialization with an object system. A detailed description of the parts of TOE will be given in the TOE Design Specification. The wrapper interface is specified in [EGK-WRP].

16 The corresponding keys and authentication data used in life cycle phase 6 are delivered securely to the Installation Agent.

17 The TOE does not include the object system, i. e. the application specific structures like the Master File (MF), the Applications, the Application Dedicated Files (ADF), the Dedicated Files (DF), Elementary Files (EF) and internal security objects including TSF data.

18 The TOE and the installed application specific object system build a smart card product, like an electronic Health Card (eHC), a Professional Health Card (eHPC) or a Secure Module Card of Type B, K or KT (SMC) according to Specifications referred in [EGK-COS, E.5.1]. This smart card product is delivered to the end-user (Personalization Agent).

19 In this ST the antenna itself is not considered as part of the TOE. Therefore the antenna integration may appear during manufacturing as well as after TOE's delivery. In case the antenna integration is part of TOE manufacturing it will be considered in the ALC documentation.

20 The Guidance documentation provides further requirements for the manufacturer and security measures required for protection of the TOE until reception by the end-user.

21 TOE's security features including authentication, access control, key management, cryptographic support, TSF data management, export of non-confidential TSF data of the object system will be described in more details in the following section.

1.4.2 TOE security features for operational use

22 The export of non-confidential TSF data of the object systems supports verification of correct implementation of the object system of the smart card during manufacturing and testing. The exported TSF data include all security attributes of the objects system as a whole and of all objects but excludes any confidential authentication data. The wrapper provides communication interfaces between the COS and a verification tool (cf. [EGK-WRP]). The verification tool sends commands for the COS through the wrapper. The wrapper encodes the data in a standardized format for the export to the verification tool. The verification tool compares the response of the smart card with the object system definition. For details refer to the Administrator's Guidance [TCOSGD].

23 The security attributes of human users are associated with password objects. The human user selects the password object and therefore the role gained by the subject acting for this human user after successful authentication. 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 select a new password and to make this password and this role effective (the *transportStatus* changes to *regularPassword*). Note that different password objects may be associated with the same role.

24 The PUC defined for the attribute *secret* is intended for password management and the authorization gained by successful authentication is limited to reset of the *retryCounter* and setting a new *secret*.

25 The physical part of the smartcard containing the IC may be protected by additional physical security measures (e.g. watermark, security printing) which bind the TOE to legitimate smartcard holder. This is not an authentication feature provided by the TOE.

26 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 and therefore the role gained by the subject acting for this device after successful authentication. A device may be also associated with a certificate containing a public key as authentication reference data. The authentication protocol comprises the verification of the certificate by means of a digital signature and the validation by means of a certificate chain.

27 The TOE supports access control lists for *lifeCycleStatus* values, security environments for contact based communication and for contactless communication. The TOE's access control rules contain commands defined by their class bytes and parameters.

28 The TOE supports random number generation for use by the TOE and the external world. The authentication protocols and the integrity protection of user data provided by the TOE use the hash algorithms SHA-1, SHA-256, SHA-384 and SHA-512. As message authentication code the TOE provides the non-standardized RMAC based on DES and the CMAC based on AES.

29 The protection of confidentiality, e.g. for secure messaging is supported by TDES (effective key length 168 bit[2]) and AES (key lengths 128, 192 and 256 bits). Asymmetric cryptographic algorithms implemented by the TOE are RSA (2048 and 3072 bit key lengths) for signature creation and encryption and the Elliptic Curve based algorithms EC-DH and EC-DSA for key agreement and signature creation.

30 All user specific authentication data like PIN, PUC or passwords are under full control of the legitimate card holder. It can be changed, blocked and reset depending on the life cycle phase and its status. The Initialization, Personalization and Life Cycle Management are restricted to the Administrator role and require a dedicated authentication.

31 The status and the access control rights as well as other non-confidential information on the user and TSF data and the access rules of the installed object system are provided by the TOE to the user. A detailed description of the so called "wrapper function" is given in the Administrator's Guidance [TCOSGD].

32 For further details refer to the chapter 6 "Security Requirements".

1.4.3 Non-TOE hardware/software/firmware

33 In order to be powered up and to communicate with the 'external world' the TOE needs a terminal (card reader) with contacts according to [ISO7816] or supporting the contactless communication according to [ISO14443].

34 There is no explicit non-TOE hardware, software or firmware required by the TOE to perform its claimed security features.

35 The TOE is defined to comprise the chip and the complete operating system and the wrapper tool together with the complete guidance documentation.

2 Note that the effective key length of TDES with keying option 1 is only 112 bits.

1.4.4 Life Cycle Phases Mapping

36 Following the protection profile PP0035 [PP0035, sec. 1.2.3] the life cycle phases of a smartcard can be divided into the following seven phases:

Phase 1: IC Embedded Software Development
Phase 2: IC Development
Phase 3: IC Manufacturing
Phase 4: IC Packaging
Phase 5: Composite Product Integration
Phase 6: Personalization
Phase 7: Operational Use

37 According to the PP [PPCOS] the TOE life cycle is described in terms of the following seven life cycle phases.

38 Note that the names of these life cycle phases do not match exactly the naming of the life cycle phases, which are taken over from the PP [PPCOS]. Additional information is given in the Administrator's Guidance [TCOSGD] and the ALC and AGD documentation.

Life cycle phase 1 "Smartcard embedded software development"

39 The TOE is developed in phase 1. The IC Platform Developer according to [AIS36] develops the integrated circuit, the IC Dedicated Software and the guidance documentation associated with these TOE components.

40 The software developer (i.e. the Application Developer according to [AIS36]) uses the guidance documentation for the integrated circuit and the guidance documentation for relevant parts of the IC Dedicated Software and develops the IC Embedded Software (operating system) and the guidance documentation associated with these TOE components.

41 The manufacturing documentation of the IC including the IC Dedicated Software and the Embedded Software in the non-volatile non-programmable memories (ROM) is securely delivered to the IC manufacturer. The IC Embedded Software in the non-volatile programmable memories (EEPROM) and the guidance documentation is securely delivered to the IC manufacturer.

42 This life cycle phase covers Phase 1 of [PP0035].

Life cycle phase 2 "IC development"

43 In a first step the TOE integrated circuit is produced containing the IC's Dedicated Software and the parts of the IC's Embedded Software in the non-volatile non-programmable memories (ROM). If necessary the IC manufacturer adds part of the IC Embedded Software in the non-volatile programmable memories (EEPROM).The IC manufacturer writes the IC Identification Data onto the chip to control the IC as smartcard material during the IC manufacturing and the delivery process.

44 This life cycle phase corresponds to Phase 2 of [PP0035].

Life cycle phase 3 "IC manufacturing and testing"

45 The IC manufacturer is responsible for producing the IC through three main steps: the manufacturing, testing and IC initialization.

46 This life cycle phase corresponds to Phase 3 [PP0035].

47 For the TOE only one pre-configured version of the operating system applies. The COS is completed in Phase 5. A detailed description of the sub-phases can be found in the Administrator's Guidance [TCOSGD].

48 This life cycle phase corresponds to Phase 3 of [PP0035].

Life cycle phase 4 "IC packaging and testing"

49 The IC packaging manufacturer is responsible for the IC packaging and testing.

50 This life cycle phase corresponds to Phase 4 of [PP0035] and is almost linked to the IC manufacturing phase.

Life cycle phase 5 "Smartcard product finishing process"

51 **The TOE is finished after completion and successful testing the COS by the TOE manufacturer.** Note that in this stage the TOE does not contain any object system and is therefore not ready yet for the end-use phase.

52 The TOE is delivered as a chip with a completed COS.

53 The keys and authentication data (the FORMAT APDUs) for opening phase 6 is delivered securely to the Installation Agent.

54 The TOE may be already integrated in a smart card. In this case the Card Manufacturer acts before TOE's delivery and the phase 5 is closed after completion. The antenna integration is part of the production process and is therefore subject to auditing.

55 The TOE's chip can also be delivered as a module that will be installed later in a smart card. Note that since in this ST the antenna is not considered as part of the TOE (cf. para. 19), there is no impact on TOE's delivery as a module.

56 The completion procedure is made by the Completion Agent, who finishes the TOE. This phase includes the COS testing.

57 If the TOE is completed as a module, it will be delivered to the Card Manufacturer only. The TOE will be integrated in a smart card and is delivered back to the Completion Agent. This is considered also as part of phase 5. The Card Manufacturer finishes the card production, including antenna installation, with the ready-made TOE. This second part of this phase is a usage of the TOE in a controlled environment covered by the guidance documentation.

58 After closing this phase the TOE is ready for installing an Object System (Installation) followed by the import User Data (Personalization).

59 This life cycle phase corresponds to Phase 5 of [PP0035].

Life cycle phase 6 "Smartcard personalization"

60 There are two user roles (Installation and Personalization Agent) foreseen in this phase, which are identified by corresponding authentication data (FORMAT APDUs). These roles may merge in a single instance, but nevertheless they are clearly different. The Installation Agent is able and is responsible for the authentication data used by the Personalization Agent.

61 The keys and authentication data (the FORMAT APDU) for the Personalization procedure is delivered securely from the Installation Agent to the Personalization Agent if these roles are assigned to different subjects.

62 The Personalization with User Data, e.g. card holder identification data, may be separated from the personalization of the TOE as an SSCD, e.g. the generation of a signature key.

63 *Application Note 1*: Note also that from a hardware point of view this cycle phase is already an operational use of the composite product and no more a personalization of the hardware. The hardware's "Personalization" (cf. [HWST]) ends with the completion of the TOE and should not be confused with the Personalization described in the Administrator Guidance [TCOSGD].

64 This life cycle phase corresponds to Phase 6 of [PP0035].

Life cycle phase 7 "Smartcard end-usage"

65 The TOE is used by the card holder corresponding to the implemented object system. The user data can be read according to the access rules of the object system.

66 This life cycle phase corresponds to the Phase 7 of the [PP0035].

67 The security environment for the TOE and the ST of the underlying platform match, the Phases up to 6 are covered by a controlled environment as required in [HWCR, p. 41]. In Phase 7 (Operational Use) no restrictions apply.

1.4.5 TOE Boundaries

1.4.5.1 TOE Physical Boundaries

68 Smart card as used in this ST means an integrated circuit containing a microprocessor, (CPU), a coprocessor for special (cryptographic) operations, a random number generator, volatile and non-volatile memory, and associated software, packaged and embedded in a carrier. The integrated circuit is a single chip incorporating CPU and memory which include RAM, ROM, and EEPROM.

69 The chip is embedded in a module which provides the capability for standardized connection to systems separate from the chip through TOE's interfaces in accordance with ISO standards.

70 The physical constituent of the TOE is the initialized chip with an operating system in ROM and EEPROM only and without any object system.

71 After the Installation of an object system the TOE can be personalized for the end-usage phase as, e.g. an electronic Health Card.

1.4.5.2 TOE Logical Boundaries

72 All card accepting devices (Host Applications) will communicate through the I/O interface of the operating system by sending and receiving octet strings. The logical boundaries of the TOE are given by the complete set of commands of the TCOS operating system for access, reading, writing, updating or erasing data.

73 The input to the TOE is transmitted over the physical interface as an octet string that has the structure of Command Application Protocol Data Unit (CAPDU). The output octet string from the TOE has the structure of a Response Application Protocol Data Unit (RAPDU).

74 The Application Protocol Data Units or TCOS commands that can be used in the operating systems are described in more detail in another document.

2 Conformance Claim

2.1 CC Conformance Claims

75 This Security Target claims conformance to Common Criteria for Information Technology Security Evaluation [CC],

Part 1: Introduction and general model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012,

Part 2: Security functional components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012,

Part 3: Security assurance components; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

as follows:

Part 2 extended, Part 3 conformant.

76 The Common Methodology for Information Technology Security Evaluation, Evaluation methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012, [CC] has to be taken into account. The evaluation follows the Common Evaluation Methodology (CEM) with current final interpretations.

2.2 PP Claims

77 This ST claims *strict* conformance to 'Common Criteria Protection Profile Card Operating System Generation 2 (PP COS G2), BSI-CC-PP-0082-V2-2014, version 1.9' [PPCOS].

2.3 Package Claims

78 The optional packages ("Crypto Box", "Contactless" and "Logical Channel") are selected and implemented by the TOE.

79 The evaluation of the TOE is a composite evaluation and uses the results of the CC evaluation provided by [HWCR]. The IC hardware platform and its primary embedded software are evaluated at level EAL 5.

80 The evaluation assurance level of the TOE is EAL4 augmented with ALC_DVS.2, ATE_\ DPT.2 and AVA_VAN.5 as defined in [CC][3].

2.4 Conformance Claim Rationale

81 The TOE type is a smartcard which is consistent with the TOE type of the claimed PP.

82 The following Security Problem Definition chapter, the Security Objectives and the Security Requirements are taken over completely from the claimed PP.

[3] In this ST the backslash provides line breaks for CC conformant identifiers. It should not be considered as a part of the identifier. Identifiers containing natural words are hyphenated as usual.

83 All the objectives, security policies (if applicable) and security requirements from the selected packages are integrated in the corresponding sections taken over from main part of the PP.

84 The optional package “PACE for Proximity Coupling Device” is not selected because it is not implemented by the TOE.

85 The Conformance Claim rationale for the Security IC Platform PP [PP0035] is given already in the Protection Profile [PPCOS] and will not repeated here.

3 Security Problem Definition

3.1 Assets and External Entities

86 As defined in section 1.3 the TOE is a smart card platform implementing the Card Operating System (COS) according [EGK-COS] without any object system. In sense of the BSI-CC-PP-0035-2007 [PP0035] the COS is User Data and Security IC Embedded Software.

87 The primary assets are User Data to be protected by the COS as long as they are in scope of the TOE and the security services provided by the TOE (please refer to the COS Specification [EGK-COS] for the term definitions).

Asset	Definition
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.

Table 1: Primary assets

88 Elementary files (EF) may be stored in the MF, any DF, or Application and 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 (cf. Tables 10, 11 and 12).

89 The protection profile for the COS [PPCOS] considers the following external entities:

External entity	Definition
World	Any user independent on identification or successful authentication[4]
Human User	The person authenticated by password or PUC
Device	An external device authenticated by cryptographic operation
Device with contactless communication	An external Device communicating with the TOE through the contactless interface. The subject bind to this device has the security attribute “*kontaktlos*” (contactless communication) (added by Package contactless).
Device authenticated using PACE protocol in PCD role	An external Device communicating with the TOE through the contactless interface and successful authenticated by PACE protocol in PCD role (added by Package contactless).

Table 2: External Entities[5]

[4] The user World corresponds to the access condition ALWAYS in [EGK-COS]. An authenticated Human User or Device is allowed to use the right assigned for World.

[5] This table defines external entities and subjects in the sense of [CC]. Subjects can be recognized by the TOE independent of their nature (human or technical user). As result of an appropriate identification and authentication process, the TOE creates

3.2 Threats

90 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 stored in or protected by the TOE and the method of TOE's use in the operational environment.

91 The following threats are defined in the Protection Profile [PP0035]: T.Leak-Inherent, T.Phys-Probing, T.Malfunction, T.Phys-Manipulation, T.Leak-Forced, T.Abuse-Func, T.RND. All threats are part of the Protection Profile [PPCOS] and are taken over into this ST. The following table lists all these threats with the corresponding reference.

Threat name	Short description	Reference to para in [PP0035]
T.Leak-Inherent	Inherent Information Leakage	78
T.Phys-Probing	Physical Probing	79
T.Malfunction	Malfunction due to Environmental Stress	80
T.Phys-Manipulation	Physical Manipulation	81
T.Leak-Forced	Forced Information Leakage	82
T.Abuse-Func	Abuse of Functionality	83
T.RND	Deficiency of Random Numbers	84

Table 3: Threats defined in BSI-CC-PP-0035-2007 and taken over into this ST

92 Please refer to [PP0035] for further descriptions and the details.

93 The TOE shall avert the threat "Forge of User or TSF data (T.Forge_Internal_Data)" as specified below.

T.Forge_Internal_Data Forge of User or TSF data

94 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.

95 The TOE shall avert the threat "Compromise of confidential User or TSF data (T.Compromise_Internal_Data)" as specified below.

T.Compromise_Internal_Data Compromise of confidential User or TSF data

96 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.

97 The TOE shall avert the threat "Misuse of TOE functions (T.Misuse)" as specified below.

– for each of the respective external entity – an 'image' inside and 'works' then with this TOE internal image (also called subject in [CC]). From this point of view, the TOE itself perceives only 'subjects' and, for them, does not differ between 'subjects' and 'external entities'. There is no dedicated subject with the role 'attacker' within the current security policy, whereby an attacker might 'capture' any subject role recognized by the TOE.

T.Misuse Misuse of TOE functions

98 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.

99 The TOE shall avert the threat “Malicious Application (T.Malicious_Application)” as specified below.

T.Malicious_Application Malicious Application

100 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.

101 The TOE shall avert the threat “Cryptographic attack against the implementation (T.Crypto)” as specified below.

T.Crypto Cryptographic attack against the implementation

102 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.

103 The TOE shall avert the threat “Interception of Communication (T.Intercept)” as specified below.

T.Intercept Interception of Communication

104 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.

105 The TOE shall avert the threat “Wrong Access Rights for User Data or TSF Data (T.WrongRights)” as specified below.

T.WrongRights Wrong Access Rights for User Data or TSF Data

106 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.

3.3 Organizational Security Policies

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

108 The following OSP is defined in the Protection Profile [PP0035]. This OSP is part of the Protection Profile and is taken over into this ST. Please refer to [PP0035] for further descriptions and the details.

OSP name	Short description	Reference to para in [PP0035]
P.Process-TOE	Protection during TOE Development and Production	86

Table 4: Overview of OSP in BSI-CC-PP-0035-2007 and taken over into this ST

109 The following OSP is defined in the Logical channel Package:

OSP.Logicalchannel Logical channel

110 The TOE supports and the operational environment uses logical channels bound to independent subjects.

111 *Application Note 2:* The COS specification [EGK-COS] describes the concept of logical channels in chapter 12.

3.4 Assumptions

112 The assumptions describe the security aspects of the environment in which the TOE will be used or is intended to be used.

113 The assumptions A.Process-Sec-IC, A.Plat-Appl and A.Resp-Appl defined in the Protection Profile [PP0035] address the operational environment of the Security IC, i.e. the COS part of the current TOE and the operational environment of the current TOE. The aspects of these assumptions relevant for the COS part of the current TOE address the development process of the COS and evaluated according to composite evaluation approach. Therefore these assumptions are refined in the PP [PPCOS] in order to address the assumptions about the operational environment of the current TOE. The following table lists and maps these security assumptions for the operational environment with the corresponding reference.

Assumptions defined in [PP0035]	Reference to para in [PP0035]	Refined assumptions for the operational environment of the current TOE	Rationale for the changes
A.Process-Sec-IC	91	A.Process-Sec-SC	While the TOE of BSI-CC-PP-0035-2007 is delivered after Phase 3 “IC Manufacturing and Testing” or Phase 4 “IC Packaging” the current TOE is delivered after Phase 5 “Composite Product Integration” and before Phase 6 “Personalization”. The protection during Phase 4 may and during Phase 5 shall be addressed by security of the development environment of the current TOE. Only protection during Personalization is in responsibility of the operational environment.
A.Plat-Appl	93	removed	Usage of Hardware Platform as TOE of PP-0035 addressed by A.Plat-Appl is covered by ADV class related to COS as part of the current TOE.
A.Resp-Appl	95	A.Resp-ObjS	The user data of the TOE of BSI-CC-PP-0035-2007 are the Security IC Embedded Software, i.e. the COS part of the TOE, the TSF data of the current TOE and the user data of

Assumptions defined in [PP0035]	Reference to para in [PP0035]	Refined assumptions for the operational environment of the current TOE	Rationale for the changes
			the COS. The object system contains the TSF data and defines the security attributes of the user data of the current TOE.

Table 5: Overview of assumptions defined in BSI-CC-PP-0035-2007 and implemented by the TOE

114 The developer of applications for COS must ensure the appropriate “Protection during Packaging, Finishing and Personalization (A.Process-Sec-SC)” while developing the application.

A.Process-Sec-SC Protection during Personalisation

115 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).

116 *Application Note 3*: If the role of the Personalization Agent is not assigned to the same subject as the Installation Agent, the Installation Agent is responsible for the quality of key used for authentication of the Personalization Agent.

117 The developer of applications for COS must ensure the appropriate “Usage of COS (A.Plat-COS)” while developing the application.

A.Plat-COS Usage of COS

118 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.

119 The developer of applications for COS must ensure the appropriate “Treatment of User Data by the Object System (A.Resp-ObjS)” while developing the application.

A.Resp-ObjS Treatment of User Data by the Object System

120 All User Data and TSF Data of the TOE are treated in the object system as defined for its specific application context.

4 Security Objectives

121 This chapter describes the security objectives for the TOE and the security objectives for the TOE environment.

4.1 Security Objectives for the TOE

122 The following TOE security objectives address the protection provided by the TOE *independent* of the TOE environment.

123 The following Security Objectives for the TOE are defined in the Protection Profile [PP0035]. The Security Objectives for the TOE are part of this Protection Profile and are taken over into this PP. Please refer to BSI-CC-PP-0035-2007 for further descriptions and the details. The following table lists all Security Objectives taken over with the corresponding reference.

Security Objective's name	Short description	Reference to para in [PP0035]
O.Leak-Inherent	Protection against Inherent Information Leakage	100
O.Phys-Probing	Protection against Physical Probing	101
O.Malfunction	Protection against Malfunctions	102
O.Phys-Manipulation	Protection against Physical Manipulation	103
O.Leak-Forced	Protection against Forced Information Leakage	104
O.Abuse-Func	Protection against Abuse of Functionality	105
O.Identification	TOE Identification	106
O.RND	Random Numbers	107

Table 6: Overview of Security Objectives for the TOE defined in BSI-CC-PP-0035-2007 and taken over into this ST

124 Please refer to [PP0035] for further descriptions and the details.

125 Additionally the following Security Objectives for the TOE are defined:

126 The TOE shall provide "Integrity of internal data (O.Integrity)" as specified below.

O.Integrity Integrity of internal data

127 The TOE must ensure the integrity of the User Data, the security services and the TSF data under the TSF scope of control.

128 The TOE shall provide "Confidentiality of internal data (O.Confidentiality)" as specified below.

O.Confidentiality Confidentiality of internal data

129 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.

130 The TOE shall provide a "Treatment of User and TSF Data (O.Resp-COS)" as specified below.

O.Resp-COS Treatment of User and TSF Data

131 The User Data and TSF data (especially cryptographic keys) are treated by the COS as defined by the TSF data of the object system.

132 The TOE shall provide “Support of TSF data export (O.TSFDataExport)” as specified below.

O.TSFDataExport Support of TSF data export

133 The TOE must provide correct export of TSF data of the object system excluding confidential TSF data for external review.

134 The TOE shall provide “Authentication of external entities (O.Authentication)” as specified below.

O.Authentication Authentication of external entities

135 The TOE supports the authentication of human users and external devices. The TOE is able to authenticate itself to external entities.

136 The TOE shall provide “Access Control for Objects (O.AccessControl)” as specified below.

O.AccessControl Access control for objects

137 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.

138 The TOE shall provide “Generation and import of keys (O.KeyManagement)” as specified below.

O.KeyManagement Generation and import of keys

139 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.

140 The TOE shall provide “Cryptographic functions (O.Crypto)” as specified below.

O.Crypto Cryptographic functions

141 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.

142 The TOE shall provide a “Secure messaging (O.SecureMessaging)” as specified below.

O.SecureMessaging Secure messaging

143 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.

144 The TOE shall provide a “Trusted channel (O.Trustedchannel)” as specified below (this is an objective from the Crypto Box package).

O.Trustedchannel Trusted channel

145 The TOE supports trusted channel for protection of the confidentiality and the integrity for commands to be sent to successful authenticated device and receiving responses from this device on demand of the external application.

146 The TOE shall provide a “Protection of contactless communication with PACE (O.PACE_CHIP)” as specified below (this is an objective from the Package Contactless).

O.PACE_CHIP Protection of contactless communication with PACE/PICC

147 The TOE supports the chip part of the PACE protocol in order to protect the confidentiality and the integrity of data communicated through the contactless interface of the TOE.

148 The TOE shall provide a “Support of more than one logical channel (O.Logicalchannel)” as specified below (this is an objective from the Logical channel Package).

O.Logicalchannel Support of more than one logical channel

149 The TOE supports more than one logical channel each bound to an independent subject.

4.2 Security Objectives for the Operational Environment

150 This section describes the security objectives for the operational environment enforced by the Security IC Embedded Software.

151 The following security objectives for the operational environment of the security IC are defined in the Protection Profile [PP0035]. The operational environment of the Security IC as TOE in BSI-CC-PP-0035-2007 comprises the COS part of the current TOE and the operational environment of the current TOE. Therefore these security objectives of the operational environment are split and refined. The aspects relevant for the COS part of the current TOE shall be fulfilled in the development process of the COS and evaluated according to composite evaluation approach. The remaining aspects of the security objectives for the operational environment defined in BSI-CC-PP-0035-2007 are addressed in new security objectives for the operational environment of the current PP. The following table lists and maps these security objectives for the operational environment with the corresponding reference.

Security Objectives for the operational environment defined in [PP0035	Reference to para in [PP0035]	Refined security objectives for the operational environment of the current TOE	Rationale of the changes
OE.Plat-Appl	109	removed	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.
OE.Resp-Appl	110	OE.Resp-ObjS	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.

Security Objectives for the operational environment defined in [PP0035	Reference to para in [PP0035]	Refined security objectives for the operational environment of the current TOE	Rationale of the changes
			This objective shall be ensured by the TOE and the object system.
OE.Process-Sec-IC	111	OE.Process-Card	The policy defined for the Security platform IC is extended to the current TOE.

Table 7: Overview of Security Objectives for the Operational Environment defined in BSI-CC-PP-0035-2007 and taken over into this ST

152 Please refer to [PP0035] for further descriptions and the details.

153 The Security IC Embedded Software shall provide “Usage of COS (OE.Plat-COS)” as specified below

OE.Plat-COS Usage of COS

154 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.

155 The Security IC Embedded Software shall provide “Treatment of User Data (OE.Resp-ObjS)” as specified below.

OE.Resp-ObjS Treatment of User Data

156 All User Data and TSF Data of the object system are defined as required by the security needs of the specific application context.

157 The operational environment of the TOE shall provide “Protection of Card during Personalization (OE.Process-Card)” as specified below

OE.Process-Card Protection of Smartcard during Personalization

158 Security procedures shall be used after delivery of the TOE during Phase 6 Smartcard personalization 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, unauthorized personalization or unauthorized use.

159 The operational environment of the TOE shall provide “Secure messaging support of external devices (OE.SecureMessaging)” as specified below (this is an objective from the Crypto Box package).

OE.SecureMessaging Secure messaging support of external devices

160 The external device communicating with the TOE through a trusted channel supports device authentication with key derivation, secure messaging for received commands and sending responses.

161 The operational environment shall provide a “PACE support by terminals (OE.PACE_Terminal)” as specified below (this is an objective from the Package Contactless).

OE.PACE_Terminal PACE support by contactless terminal

162 The external device communicating trough a contactless interface with the TOE using PACE shall support the terminal part of the PACE protocol.

163 The security objectives O.PACE_CHIP and OE.PACE_Terminal mitigate the threat T.Intercept if contactless communication between the TOE and the terminal is used and the operational environment is not able to protect the communication by other means.

164 The operational environment shall provide a “Use of logical channels (OE.Logical-channel)” as specified below (this is an objective from the Logical channel Package).

OE.Logicalchannel Use of logical channels

165 The operational environment manages logical channels bound to independent subjects for running independent processes at the same time.

166 The security objectives O.Logicalchannel and OE.Logicalchannel implement the OSP.Logicalchannel.

4.3 Security Objective Rationale

167 The following table provides an overview for security objectives coverage (TOE and its environment). It shows that all threats and OSPs are addressed by the security objectives. It also shows that all assumptions are addressed by the security objectives for the TOE environment.

	O.Identification	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging	O.Trustedchannel	O.PACE_CHIP	O.Logicalchannel	SAR ALC (IC part)	OE.Process-Sec-Card	SAR ADV (COS part)	SAR for COS part	OE.Plat-COS	OE.Resp-ObjS	OE.Process-Card	OE.SecureMessaging	OE.PACE_Terminal	OE.Logicalchannel
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																						
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	x	x									x	x	
T.WrongRights											x																				
OSP.Logicalchannel																				x										x
P.Process-TOE	x																										x			
A.Process-Sec-IC																					x	x								
A.Process-Sec-SC																						x								

	O.Identification	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging	O.Trustedchannel	O.PACE_CHIP	O.Logicalchannel	SAR ALC (IC part)	OE.Process-Sec-Card	SAR ADV (COS part)	SAR for COS part	OE.Plat-COS	OE.Resp-ObjS	OE.Process-Card	OE.SecureMessaging	OE.PACE_Terminal	OE.Logicalchannel
A.Plat-Appl	n.a.																						x							
A.Resp-Appl																								x						
A.Plat-COS																									x					
A.Resp-ObjS																										x				
A.Process-TOE																											x			

Table 8:Security Objective Rationale

168 A detailed justification required for suitability of the security objectives to coup with the security problem definition for the IC platform is given in the Protection Profile BSI-CC-PP0035 [PP0035]. For the additional threats the corresponding rationale is given in the claimed by this ST Protection Profile BSI-CC-PP0082 [PPCOS]. Hence it will not be repeated here.

169 The security objectives from the Crypto Box package O.Trustedchannel and OE.SecureMessaging mitigate the threat T.Intercept if the operational environment is not able to protect the communication by other means.

170 The security objectives O.PACE_CHIP and OE.PACE_Terminal from the package Contactless mitigate the threat T.Intercept if contactless communication is used and the operational environment is not able to protect the communication by other means.

171 The two security objectives O.Logicalchannel and OE.Logicalchannel implement the OSP.Logicalchannel.

5 Extended Components Definition

172 This Security Target uses components defined in the Protection Profile [PPCOS] as extensions to CC part 2. All these extended components are drawn from Definitions of chapter 5 of [PPCOS]. The families FAU_SAS, FCS_RNG and FMT_LIM are already defined in BSI-CC-PP0035 [PP0035]. Note that FCS_RNG is refined by [PPCOS].

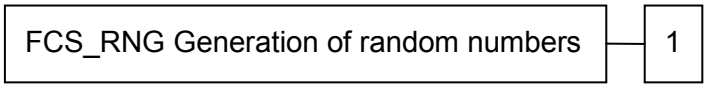
5.1 FCS_RNG Generation of random numbers

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

Family behavior

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

Component leveling:



FCS_RNG.1 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: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

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*].

5.2 FIA_API Authentication Proof of Identity

174 The family “Authentication Proof of Identity (FIA_API)” is specified as follows.

Family behavior

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 leveling:

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

Management: FIA_API.1

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

Audit: FIA_API.1

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.

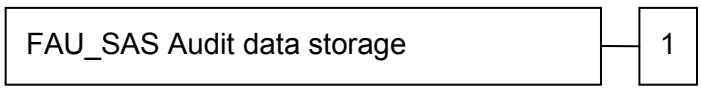
5.3 FAU_SAS Audit data storage

175 The family “Audit data storage (FAU_SAS)” is specified as follows.

Family behavior

This family defines functional requirements for the storage of audit data.

Component leveling



FAU_SAS.1 Requires the TOE to provide the possibility to store audit data.

Management: FAU_SAS.1

There are no management activities foreseen.

Audit: FAU_SAS.1

There are no actions defined to be auditable.

FAU_SAS.1 Audit storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_SAS.1.1 The TSF shall provide [assignment: *list of subjects*] with the capability to store [assignment: *list of audit information*] in the [assignment: *type of persistent memory*].

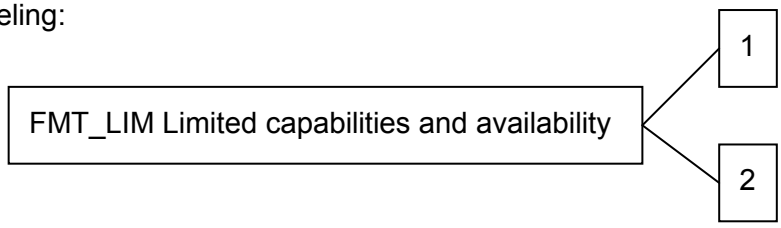
5.4 FMT_LIM Limited capabilities and availability

176 The family “Limited capabilities and availability (FMT_LIM)” is specified as follows.

Family behavior

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

Component leveling:



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

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

Management: FMT_LIM.1, FMT_LIM.2

There are no management activities foreseen.

Audit: FMT_LIM.1, FMT_LIM.2

There are no actions defined to be auditable.

The TOE Functional Requirement “Limited capabilities (FMT_LIM.1)” is specified as follows.

FMT_LIM.1 Limited capabilities

Hierarchical to: No other components.

FMT_LIM.1.1 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 [assignment: *Limited capability and availability policy*].

Dependencies: FMT_LIM.2 Limited availability.

The TOE Functional Requirement “Limited availability (FMT_LIM.2)” is specified as follows.

FMT_LIM.2 Limited availability

Hierarchical to: No other components.

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

Dependencies: FMT_LIM.1 Limited capabilities.

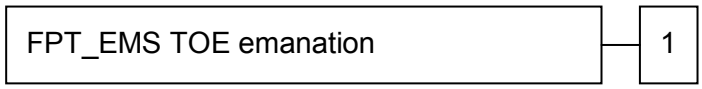
5.5 FPT_EMS TOE Emanation

177 The family “TOE Emanation (FPT_EMS)” is specified as follows.

Family behavior

This family defines requirements to mitigate intelligible emanations.

Component leveling:



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

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.

Management: FPT_EMS.1

There are no management activities foreseen.

Audit: FPT_EMS.1

There are no actions defined to be auditable.

FPT_EMS.1 TOE Emanation

Hierarchical to: No other components.

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*].

Dependencies: No other components.

5.6 FPT_ITE TSF image export

178 The family “TSF image export (FPT_ITE)” is specified as follows.

Family behavior

This family defines rules for fingerprints of TOE implementation and export of TSF data in order to allow verification of their correct implementation in the TOE. 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 specification. The exported TSF images 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 [CC].

Component leveling:

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.

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.

Dependencies: No dependencies.

FPT_ITE.2 Export of TSF data

Hierarchical to: No other components.

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.

Dependencies: No dependencies.

6 Security Requirements

179 This part of the ST defines the detailed security requirements that shall be satisfied by the TOE. The statement of **TOE security requirements** shall define the *functional* and *assurance* security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE.

180 The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in section 8.1 of Part 1 of the Common Criteria [CC]. Each of these operations is used in this ST.

181 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 are ~~crossed out~~. Refinements made by the ST author appear ***slanted, bold and underlined***.

182 The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the PP author are denoted as underlined text. Selections made by the ST author appear *slanted and underlined*.

183 The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made by the PP author are denoted by showing as underlined text. Assignments made by the ST author appear *slanted and underlined*.

184 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.

185 For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate belonging of such SFRs to same functional cluster. In such a case, the iteration operation is applied to only one single component.

6.1 Security Functional Requirements for the TOE

6.1.1 Overview

186 The following table provides an overview of security functional requirements in the context of the main security functionalities offered by the TOE:

Security Functional Group	SFR concerned
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
General Protection of User data and TSF data (section 286 and 6.1.7)	FDP_RIP.1, FDP_RIP.1/PACE.PICC, FDP_SDI.2, FPT_FLS.1, FPT_EMS.1, FPT_EMS.1/PACE.PICC, FPT_TDC.1, FPT_ITE.1, FPT_ITE.2, FPT_ITE.2/PACE, FPT_TST.1

Security Functional Group	SFR concerned
Authentication (section 6.1.4)	FIA_AFL.1/PIN, FIA_AFL.1/PUC, FIA_ATD.1, FIA_ATD.1/PACE, FIA_SOS.1, FIA_UAU.1, FIA_UAU.1/PACE, FIA_UAU.4, FIA_UAU.4/PACE.PICC, FIA_UAU.5, FIA_UAU.5/PACE.PICC, FIA_\ UAU.6, FIA_UAU.6/CB, FIA_UAU.6/PACE.PICC, FIA_API.1, FIA_API.1/CB, FMT_SMR.1, FIA_USB.1, FIA_USB.1/CB, FIA_USB.1/PACE.PICC, FIA_USB.1/LC
Access Control (section 286 and 6.1.6)	FDP_ACC.1/EF, FDP_ACF.1/EF, FDP_ACC.1/SEF, FDP_ACF.1/SEF, FDP_ACC.1/TEF, FDP_ACF.1/TEF, FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FDP_ACC.1/LC, FDP_ACF.1/LC, FDP_UCT.1/PACE, FDP_UIT.1/PACE, FMT_MSA.3, FIA_UID.1, FIA_UID.1/PACE, FMT_MSA.3/LC, FMT_SMF.1, FMT_SMR.1/PACE.PICC, 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/PACE.PICC
Cryptographic Functions (section 6.1.3)	FCS_RNG.1, FCS_RNG.1/GR, FCS_RNG.1/PACE, FCS_COP.1/SHA, FCS_COP.1/COS.3TDES, FCS_COP.1/CB.3TDES, FCS_COP.1/COS.RMAC, FCS_COP.1/CB.RMAC, FCS_CKM.1/3TDES_SM, FCS_COP.1/COS.AES, FCS_COP.1/CB.AES, FCS_COP.1/PACE.PICC.ENC, FCS_COP.1/PACE.PICC.MAC, FCS_CKM.1/AES.SM, FCS_CKM.1/RSA, FCS_CKM.1/ELC, FCS_CKM.1/DH.PACE.PICC, FCS_CKM.4/PACE.PICC, FCS_COP.1/COS.CMAC, FCS_COP.1/CB.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/CB.RSA, FCS_COP.1/COS.ELC, FCS_COP.1/CB.ELC, FCS_CKM.4
Protection of communication (section 6.1.3 and 6.1.9)	FTP_ITC.1/TC, FTP_ITC.1/PACE.PICC

Table 9: Security Functional Groups vs. SFRs

187 The SFRs related to the IC Platform are marked with the iteration /SICP as defined in the PP [PPCOS].

188 The following table provides the IC related TSF Data implemented by the TOE [PPCOS, Table 13]:

TSF Data	Definition
TOE pre-personalization data	Any data supplied by the Card Manufacturer that is injected into the non-volatile memory by the Integrated Circuits manufacturer.
TOE initialization data	Initialization Data defined by the TOE Manufacturer to identify the 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 10: IC related TSF Data

189 The security attributes of human users are stored in password objects (cf. [EGK-COS] for details). The human user selects the password object by *pwdIdentifier* 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 *pwdIdentifier*. 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*.

190 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 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.

191 The following table provides an overview of the authentication reference data and security attributes of human users and devices and the security attributes of the authentication reference data as TSF data [PPCOS, Table 14 and 15]:

User type	Authentication reference data and security attributes	Operations
Human user	**Password** Authentication reference data: *secret* Security attributes of the user role: *pwdIdentifier, transportStatus, lifeCycleStatus, flagEnabled, startSsecList* Security attributes of the secret: *interfaceDependentAccessRules, startRetryCounter, 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	Operations
Human user	**Multi-Reference password** Authentication reference data: *secret* is shared with the password identified by *pwReference*. Security attributes of the user role: *pwdIdentifier, 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: *pwdIdentifier* of the password[6], *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).
Device	**Symmetric authentication key** Authentication reference data: *macKey*[7] 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[8] *persistentCache, applicationPublicKeyList*[9] 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 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.
Device	**Secure messaging channel key**	The TOE authenticates the sender of a received com-

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

7 The symmetric authentication object contains encryption key *encKey* and a message authentication key *macKey*.

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

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

User type	Authentication reference data and security attributes	Operations
	Authentication reference data: MAC session key *SK4SM* Security attributes of *SK4SM: flagSessionEnabled* equal *SK4SM*, *Kmac* and *SSCmac*, *negotiationKeyInformation*.	mand using secure messaging.
Device	**Symmetric authentication key**	MUTUAL AUTHENTICATE, EXTERNAL AUTHENTICATE, PSO DECIPHER and PSO VERIFY CRYPTOGRAPHIC CHECKSUM used for trusted channel (added by the Package Crypto Box)
Device	**Symmetric Card Connection Object (SCCO)** Authentication reference data: SCCO stored in TOE and corresponding to the CAN, MAC session key *SK4SM* Security attributes: *keyIdentifier* of the SCCO in the *globalSecurityList* if SCCO was in MF or in *dfSpecificSecurityList* if the SCCO was in the respective folder, *SK4TC* referenced in *Kmac* and *SSCmac*	GENERAL AUTHENTICATE with (CLA,INS,P1,P2) = (x0,86,00,00) is used by TOE running PACE protocol role as PICC to authenticate the external device running PACE protocol role as PCD. (added by the Package Contactless)
TOE as PICC	*SK4SM* referenced in *macKey* and *SSCmac*	*SK4SM* is used to generate MAC for command responses. (added by the Package Contactless)

Table 11: Authentication reference data and security attributes

192 The following table defines the authentication verification data used by the TSF itself for authentication by external entities (cf. FIA_API.1) [PPCOS, Table 16]:

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, Kmac* and *SSCmac, Kenc* and *SSCenc, flagCmdEnc* and *flagRspEnc*	Responses using secure messaging The session keys are linked to the folder of the keys used by them.
TSF	**Trusted channel** Authentication verification data Session key *SK4TC* Security attributes *SK4TC* referenced in *keyReferenceList.macCalculation* and *keyReferenceList.dataEncipher*	The commands PSO COMPUTE CRYPTOGRAPHIC CHECKSUM and PSO ENCIPHER are used to generate commands received by the authenticated PICC with secure messaging. (added by the Package Crypto Box)
TSF	**Session key *SK4TC***	PSO ENCIPHER, PSO DECIPHER, PSO VERIFY CERTIFICATE and PSO VERIFY CRYPTOGRAPHIC CHECKSUM used for trusted channel (added by the Package Crypto Box)

Table 12: Authentication verification data of the TSF and security attributes

193 The COS specification associates a subject with a logical channel and its *channelContext* (cf. [EGK-COS], chapter 12). The TOE may support one subject respective logical channel or more than one independent subjects respective logical channels. The *channelContext* comprises security attributes of the subject summarized in the following table [PPCOS, Table 17]:

Security attribute	Elements	Comments
Interface		The TOE detects whether the communication uses contact based interface (value set to *kontaktbehaftet*), or contactless interface (value set to *kontaktlos*)[10]. If the TOE does not support contactless communication the TOE shall behave as *interfaceDependentAccessRules* is permanently set to *kontaktbehaftet*.
currentFolder		Identifier of the (unique) current folder
	seIdentifier	Security environment selected by means of command MANAGE SECURITY ENVIRONMENT[11]. 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.
	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
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.
	Kenc and *SSCenc*	Key for encryption and decryption and its sequence counter
	Kmac and *SSCmac*	Key for MAC calculation and verification and its sequence counter
	flagCmdEnc and *flagRspEnc*	Flags indicating encryption of data in commands respective responses

10 Note the COS specification [EGK-COS] describes this security attribute in the context of access control rules in chapter 8.1.4 only. If the TOE does not support contactless communication the document in hand shall be read assuming that this attribute is equal to "*kontaktbehaftet*".

11 Note the COS specification [EGK-COS] describes this security attribute in the informative chapter 8.8. The object system specification of the eHPC uses this security attribute for access control rules of batch signature creation.

Security attribute	Elements	Comments
	negotiationKeyInformation	*keyIdentifier* of the key used to generate the session keys and if asymmetric key was used the *accessRight* associated with this key. The *keyIdentifier* may reference to the authentication reference data used for PACE.
	accessRulesSessionkeys	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*
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.
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* 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[12].
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*.
currentFile		Identifier of the (unique) current file from *currentFolder.children*
securityStatusEvaluationCounter	*startSsec*	Must contain all values of *startSsec* and may be empty

Table 13: Security attributes of a subject

194 The following tables provide an overview of the objects, operations and security attributes defined in the PP [PPCOS, Table 18]. All references in the table refer to the technical specification of the card operating system [EGK-COS].

Object type	Security attributes	Operations
Object System	*applicationPublicKeyList, persistentCache, pointInTime*	PSO VERIFY CERTIFICATE
Folder (8.3.1)	accessRules: *lifeCycleStatus*, *shareable*, *interfaceDependentAccessRules*, *children*	SELECT, ACTIVATE, DEACTIVATE, DELETE, FINGERPRINT, GET RANDOM, LOAD APPLICATION, TERMINATE DF
Dedicated File (8.3.1.2)	Additionally to Folder: *fileIdentifier*	Identical to Folder
Application (8.3.1.1)	Additionally to Folder: *applicationIdentifier*	Identical to Folder
Application Dedicated File (8.3.1.3)	Additionally to Folder: *fileIdentifier, applicationIdentifier, children*	Identical to Folder
Elementary File (8.3.2)	*fileIdentifier*, list of *shortFileIdentifier*, *lifeCycleStatus*, *shareable* accessRules: *interfaceDependentAccessRules*, *flagTransactionMode*, *flagChecksum*	SELECT, ACTIVATE, DEACTIVATE, DELETE, TERMINATE
Transparent EF (8.3.2.1)	Additionally to Elementary File: *numberOfOctet*, *positionLogicalEndOfFile*,	Additionally to Elementary File: ERASE BINARY, READ BINARY, UPDATE BINARY,

12 The *keyIdentifier* generated by successful authentication with PACE protocol is named "*Kartenverbindungsobjekt*" in the COS specification [EGK-COS].

Object type	Security attributes	Operations
	body	WRITE BINARY
Structured EF (8.3.2.2)	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 (8.4) (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
Multi-reference Password (8.5) (MR-PIN)	*lifeCycleStatus*, *pwdIdentifier*, accessRules: *interfaceDependentAccessRules*, *startSsecList*, *flagEnabled*, *pwReference*, 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 (8.6.1)	*lifeCycleStatus*, *keyIdentifier* accessRules: *interfaceDependentAccessRules*, *encKey*, *macKey*, *numberScenario*, *algorithmIdentifier*, accessRulesSessionkeys: *interfaceDependentAccessRules*	ACTIVATE, DEACTIVATE, DELETE, TERMINATE, EXTERNAL AUTHENTICATE, GENERAL AUTHENTICATE, INTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE
Private Asymmetric Key (8.6.4)	*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 (8.6.4)	*lifeCycleStatus*, *keyIdentifier*, *oid* accessRules: *interfaceDependentAccessRules*	ACTIVATE, DEACTIVATE, DELETE, TERMINATE
Public Asymmetric Key for signature verification (8.6.4.2)	Additionally to Public Asymmetric Key: *publicRsaKey* or *publicElcKey*: *oid* *CHAT*, *expirationDate*: date	Additionally to Public Asymmetric Key: PSO VERIFY CERTIFICATE, PSO VERIFY DIGITAL SIGNATURE
Public Asymmetric Key for Authentication (8.6.4.3)	*publicRsaKey* or *publicElcKey*: *oid* *CHA*, *CHAT*, *expirationDate*: date	Additionally to Public Asymmetric Key: EXTERNAL AUTHENTICATE, GENERAL AUTHENTICATE, INTERNAL AUTHENTICATE
Public Asymmetric Key for Encryption (8.6.4.4)	Additionally to Public Asymmetric Key: *publicRsaKey* or *publicElcKey*: *oid*	Additionally to Public Asymmetric Key: PSO ENCIPHER
Card verifiable certificate (CVC) (7.1.1)	Certificate Profile Identifier (*CPI*) Certification Authority Reference (*CAR*) Certificate Holder Reference (*CHR*) Certificate Holder Autorization (*CHA*) Object Identifier (*OID*) *signature*	

Table 14: Subjects, objects, operations and security attributes

195 The TOE supports Access control lists for *lifeCycleStatus* values “Operation state (activated)”, “Operation state (deactivated)” and “Termination state”, security environments with value *seIdentifier* selected for the folder *interfaceDependentAccessRules* for contact based communication, and for *interfaceDependentAccessRules* for contactless communication.

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

197 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[13]

198 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(*pwReference*), AUT(*keyReference*), AUT(*CHA*), AUT(*CHAT*) and secure messaging conditions (cf. [EGK-COS], chapter 10.2 for details).

199 *Application Note 4:* 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.

200 The Boolean element ALWAYS provides always the Boolean value TRUE. The Boolean element NEVER provides always 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.

201 The following table gives an overview of the commands implemented by the COS. Optional commands as defined in [EGK-COS] which are not implemented by the COS are marked ~~crossed out~~.

13 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, which requires additional authentication of the signature creation application.

Operation	SFR	chapter
ACTIVATE	FMT_SMF.1, FMT_MSA.1/Life	14.2.1
ACTIVATE RECORD	FMT_SMF.1, FMT_MSA.1/SEF	14.4.1
APPEND RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF	14.4.2
CHANGE REFERENCE DATA	FIA_UAU.5, FIA_USB.1, FMT_SMF.1, FMT_MTD.1/PIN, FMT_MSA.1/PIN, FIA_AFL.1/PIN	14.6.1
~~CREATE~~	This command is optional and therefore not addressed in the SFRs.	14.2.2
DEACTIVATE	FMT_SMF.1, FMT_MSA.1/PIN	14.2.3
DEACTIVATE RECORD	FMT_SMF.1, FMT_MSA.1/SEF	14.4.3
DELETE	FIA_USB.1, FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF, FDP_ACC.1/EF, FDP_ACF.1/EF, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FMT_SMF.1, FMT_MSA.1/Life, FCS_CKM.4, FIA_USB.1/LC	14.2.4
DELETE RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF, FMT_MSA.1/SEF	14.4.4
DISABLE VERIFICATION REQUIREMENT	FMT_SMF.1, FMT_MSA.1/PIN, FIA_AFL.1/PIN, FIA_USB.1	14.6.2
ENABLE VERIFICATION REQUIREMENT	FMT_SMF.1, FMT_MSA.1/PIN, FIA_AFL.1/PIN, FIA_USB.1	14.6.3
~~ENVELOPE~~	This command is optional and therefore not addressed in the SFRs.	14.9.1
ERASE BINARY	FDP_ACC.1/TEF, FDP_ACF.1/TEF	14.3.1
ERASE RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF, FMT_MSA.1/SEF	14.4.5
EXTERNAL AUTHENTICATE	FIA_UAU.4, FIA_UAU.5, FIA_USB.1, FIA_USB.1/CB, FCS_RNG.1, FCS_CKM.1/AES.SM, FCS_COP.1/COS.RSA.V, FCS_COP.1/COS.ECDSA.V, FCS_COP.1/CB.3TDES, FCS_COP.1/CB.RMAC, FCS_COP.1/CB.AES, FCS_COP.1/CB.CMAC	14.7.1
FINGERPRINT	FPT_ITE.1, FDP_ACF.1/MF_DF	14.9.2
GENERAL AUTHENTICATE	FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_UAU.6/CB, FIA_API.1, FIA_API.1/CB, FIA_USB.1, FIA_USB.1/CB, FCS_RNG.1, FCS_COP.1/COS.AES, FCS_CKM.1/AES.SM, FIA_UAU.5/PACE.PICC, FIA_UAU.6/PACE.PICC, FIA_USB.1/PACE.PICC	14.7.2
GENERATE ASYMMETRIC KEY PAIR	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FMT_SMF.1, FCS_CKM.1/RSA, FCS_CKM.1/ELC	14.9.3
GET CHALLENGE	FCS_RNG.1	14.9.4
~~GET DATA~~	This command is optional and therefore not addressed in the SFRs.	14.5.1.
GET PIN STATUS	FMT_SMF.1, FMT_MSA.1/PIN	14.6.4
GET RANDOM	FCS_RNG.1, FCS_RNG.1/GR	14.9.5
~~GET RESPONSE~~	This command is optional and therefore not addressed in the SFRs.	14.9.6
GET SECURITY STATUS KEY	FMT_SMF.1, FMT_MSA.1/Auth	14.7.3
INTERNAL AUTHENTICATE	FIA_API.1,FIA_API.1/CB, FCS_CKM.1/AES.SM, FCS_COP.1/COS.RSA.S, FCS_COP.1/COS.ECDSA.S, FCS_COP.1/CB.3TDES, FCS_COP.1/ CB.RMAC, FCS_COP.1/CB.AES, FCS_COP.1/CB.CMAC	14.7.4
LOAD APPLICATION	FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF, FMT_SMF.1, FMT_MSA.1/Life	14.2.5
LIST PUBLIC KEY	FPT_ITE.2, FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF	14.9.7
MANAGE CHANNEL	FIA_UID.1, FIA_UAU.1, FIA_USB.1/LC, FMT_MSA.3	14.9.8
MANAGE SECURITY ENVIRON-	FIA_USB.1, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3	14.9.9

Operation	SFR	chapter
MENT		
MUTUAL AUTHENTICATE	FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_UAU.6/CB, FIA_API.1, FIA_\ API.1/CB, FIA_USB.1, FIA_USB.1/CB, FCS_RNG.1, FCS_CKM.1/AES.SM, FCS_COP.1/CB.3TDES, FCS_COP.1/CB.RMAC, FCS_COP.1/CB.AES, FCS_COP.1/CB.CMAC	14.7.1
PSO COMPUTE CRYPTOGRAPH-IC CHECKSUM	FIA_API.1, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FCS_COP.1/COS.CMAC, FCS_COP.1/CB.RMAC, FCS_COP.1/CB.CMAC, FIA_UAU.5/PACE, FIA_UAU.6/PACE.PICC, FIA_USB.1/PACE	14.8.1
PSO COMPUTE DIGITAL SIGNA-TURE, without "message recovery"	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.RSA.S, FCS_COP.1/COS.ECDSA.S	14.8.2.1
PSO COMPUTE DIGITAL SIGNA-TURE, with "message reco-very"	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.ECDSA.S	14.8.2.2
PSO DECIPHER	FIA_USB.1 ,FIA_USB.1/CB, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.RSA, FCS_COP.1/COS.ELC, FCS_COP.1/CB.3TDES, FCS_COP.1/CB.AES, FIA_UAU.5/PACE.PICC, FIA_UAU.6/CB, FIA_UAU.6/PACE.PICC, FIA_USB.1/PACE.PICC	14.8.3
PSO ENCIPHER	FIA_API.1, FIA_API.1/CB, FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_\ MSA.3, FCS_COP.1/COS.RSA, FCS_COP.1/COS.ELC, FCS_COP.1/CB.\ 3TDES, FCS_COP.1/CB.AES, FCS_COP.1/CB.RSA, FCS_COP.1/CB.ELC	14.8.4
~~PSO HASH~~	This command is optional and therefore not addressed in the SFRs.	-
PSO TRANSCIPHER using RSA	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.RSA,FCS_COP.1/COS.ELC	14.8.6.1
PSO TRANSCIPHER using ELC	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.RSA, FCS_COP.1/COS.ELC	14.8.6.3
PSO VERIFY CERTIFICATE	FMT_SMF.1, FMT_MTD.1/Auth, FCS_COP.1/COS.RSA.V, FCS_COP.1/COS.ECDSA.V, FDP_ACC.1/KEY, FDP_ACF.1/KEY	14.8.7
PSO VERIFY CRYPTOGRAPHIC CHECKSUM	FIA_USB.1, FIA_USB.1/CB, FDP_ACC.1/KEY, FDP_ACF.1/KEY, CS_COP.1/COS.RMAC, FCS_COP.1/COS.CMAC, FCS_COP.1/CB.CMAC	14.8.8
PSO VERIFY DIGITAL SIGNA-TURE	FDP_ACC.1/KEY, FDP_ACF.1/KEY, FMT_MSA.3, FCS_COP.1/COS.ECDSA.V	14.8.9
~~PUT DATA~~	This command is optional and therefore not addressed in the SFRs.	14.5.2
READ BINARY	FDP_ACC.1/TEF, FDP_ACF.1/TEF	14.3.2
READ RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF	14.4.6
RESET RETRY COUNTER	FIA_AFL.1/PUC, FIA_UAU.5, FMT_SMF.1, FMT_MTD.1/PIN, FMT_MSA.1/PIN	14.6.5
~~SEARCH BINARY~~	This command is optional and therefore not addressed in the SFRs.	14.3.3
SEARCH RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF	14.4.7
SELECT	FIA_USB.1, FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF, FDP_ACC.1/EF, FDP_ACF.1/EF	14.2.6
SET LOGICAL EOF	FDP_ACC.1/TEF, FDP_ACF.1/TEF	14.3.4
TERMINATE	FMT_SMF.1, FMT_MSA.1/Life	14.2.9
TERMINATE CARD USAGE	FMT_SMF.1, FMT_MSA.1/Life	14.2.7
TERMINATE DF	FMT_SMF.1, FMT_MSA.1/Life	14.2.8
UPDATE BINARY	FDP_ACC.1/TEF, FDP_ACF.1/TEF	14.3.5

Operation	SFR	chapter
UPDATE RECORD	FDP_ACC.1/SEF, FDP_ACF.1/SEF	14.4.8
VERIFY	FIA_AFL.1/PIN, FIA_UAU.5, FIA_USB.1, FMT_SMF.1, FMT_MSA.1/PIN	14.6.6
WRITE BINARY	FDP_ACC.1/TEF, FDP_ACF.1/TEF	14.3.6
~~WRITE RECORD~~	This command is optional and therefore not addressed in the SFRs.	14.4.9

Table 15: Mapping between commands described in COS specification [EGK-COS] and the SFR

202 There are only two additional commands provided by the TOE:

Operation	SFR
FORMAT	FMT_SMF.1, FMT_SMR.1
GET CARD INFO	FPT_ITE.2, FMT_MTD.1/NE

Table 16: Mapping between additional commands provided by the TOE and the SFR

203 *Application Note 5:* The command FORMAT is a management command available only in Life Cycle Phases 5 and 6. It is used for installation of the COS already before the TOE is finished, later it is used for transition control in the Life Cycle Phases. After the Phase 6 (Personalization) this command is no more available. Note that the FORMAT command requires strong user authentication for the Initialization and for the Personalization (cf. FMT_SMR.1 on p. 91). It is bound to a restricted usage counter.

204 *Application Note 6:* The command GET CARD INFO provided by the TOE is used by the wrapper tool. It provides only public information and is available also after Life Cycle Phase 6.

205 All SFRs from section 6.1 ”Security Functional Requirements for the TOE” of the BSI-CC-PP-0035-2007 [PP0035] are part of the BSI-CC-PP0082 [PPCOS]. 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 corresponding SFR name from BSI-CC-PP-0035-2007. For further descriptions, details, and interpretations refer to [PP0035]:

- 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

6.1.2 Class FAU Security Audit

206 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[14] with the capability to store the Initialization Data and/or Pre-Personalization Data and/or supplements of the Security IC Embedded Software[15] in the *not changeable configuration page area and non-volatile memory*[16].

6.1.3 Class FCS Cryptographic Support

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

(1) key length 256 bit
   a. brainpoolP256r1 defined in RFC5639 [RFC5639],
   b. ansix9p256r1 defined in ANSI X.9.62, identical to P-256 defined in [FIPS186],

(2) key length 384 bit
   a. brainpoolP384r1 defined in RFC5639 [RFC5639],
   b. ansix9p384r1 defined in ANSI X.9.62, identical to P-384 defined in [FIPS186],

(3) key length 512 bit
   a. brainpoolP512r1] defined in RFC5639 [RFC5639].

208 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 bit modulus length.

209 FCS_RNG.1/SICP Random number generation (HW)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1/ SICP	The TSF shall provide a *physical*[17] random number generator **of class PTG.2**[18] that implements[19] (PTG.2.1) A total failure test detects a total failure of entropy source

14 [assignment: *list of subjects*]

15 [assignment: *list of audit information*]

16 [assignment: *type of persistent memory*]

17 [selection: *deterministic, hybrid deterministic, physical, hybrid physical*]

18 [selection: ***DRG.3, DRG.4, PTG.2, PTG.3***]

19 [assignment: *list of security capabilities* ***of the selected RNG class***]

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/ SICP The TSF shall provide *numbers in the format 8- or 16-bit* that meet[20]

(PTG.2.6) Test procedure A, as defined in [6] 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.

210 *Application Note 7:* This is the functional requirement FCS_RNG.1 fulfilled by the Hardware TOE and taken over from the hardware ST [HWST].

211 FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *hybrid deterministic*[21] random number generator ***DRG.4***[22] that implements[23]

(DRG.4.1) The internal state of the RNG shall *use PTRNG of class PTG.2 as random source*[24].

(DRG.4.2) The RNG provides forward secrecy.

(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.

20 [assignment: *a defined quality metric*]

21 [selection: *deterministic, hybrid deterministic, physical, hybrid physical*]

22 [selection: ***DRG.3, DRG.4, PTG.2, PTG.3***]

23 [assignment: *list of security capabilities* ***of the selected RNG class***]

24 [selection: *use PTRNG of class PTG.2 as random source, have* [assignment: *work factor*], *require* [assignment: *guess work*]]

(DRG.4.4) The RNG provides enhanced forward secrecy *on condition "session closed or aborted"*[25].

(DRG.4.5) The internal state of the RNG is seeded by *a PTRNG of class PTG.2*[26].

FCS_RNG.1.2 The TSF shall provide random numbers that meet[27]

(DRG.4.6) The RNG generates output for which $k > 2^{34}$ strings[28] of bit length 128 are mutually different with probability $1-\varepsilon$, with $\varepsilon < 2^{-16}$.

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A[29], *the NIST and the dieharder*[30] *tests*[31].

212 *Application Note 8:* This SFR requires the TOE to generate random numbers used for key generation according to TR-03116-1 [TR3116-1, section 3.4], requiring RNG classes identified in the selection in element FCS_RNG.1.1 and recommending RNG of class PTG.3. Note that the RNG of class DRG.4 are hybrid deterministic and of class PTG.3 are hybrid physical which are not addressed in BSI-CC-PP-0035. The implementation of a physical RNG used for PACE requires the class PTG.3 (cf. [TR3116-1, sec. 3.4]), which does not exclude the selection DRG.4 made in this ST.

213 The COS specification [EGK-COS] requires to implement RNG for

- the command GET CHALLENGE,
- the command GET RANDOM,
- the authentication protocols as required by FIA_UAU.4,
- the key agreement for secure messaging
- according to TR-03116 [TR3116-1, section 3.4].

214 FCS_RNG.1/GR Random number generation – GET RANDOM command

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1/GR The TSF shall provide a *hybrid physical*[32] random number generator *PTG.3*[33] **for GET RANDOM** that that implements[34]

---

25 [selection: *on demand, on condition* [assignment: *condition*], *after* [assignment: *time*]]

26 [selection: *internal entropy source*, *PTRNG of class PTG.2*, *PTRNG of class PTG.3*, [other selection]]

27 [assignment: *a defined quality metric*]

28 [assignment: *number of strings*]

29 [assignment: *additional test suites*]

30 The selected here test suites http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.1.zip and http://www.phy.duke.edu/~rgb/General/dieharder/dieharder-3.31.0.tgz are available at NIST and Dieharder web sites. Note that the dieharder tests include Marsaglia's "Diehard battery of tests" and NIST tests.

31 [assignment: *additional test suites*]

32 [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]

33 [selection: ***PTG.2, PTG.3***]

34 [assignment: *list of security capabilities* ***of the selected RNG class***]

(PTG.3.1) 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.3.2) 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*[35].

(PTG.3.3) 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 and the seeding of the DRG.3 post-processing algorithm have been finished successfully or when a defect has been detected.

(PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.3.5) The online test procedure checks the quality of the raw random number sequence. It is triggered *continuously*[36]. 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.

(PTG.3.6) The algorithmic post-processing algorithm belongs to Class DRG.3 with cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm shall not exceed its input data rate.

FCS_RNG.1.2/GR The TSF shall provide random ~~numbers~~ *octets of bits*[37] that meet[38]

(PTG.3.7) Statistical test suites cannot practically distinguish the internal random numbers form output sequences of an ideal RNG. The internal random numbers must pass test procedure A[39], *the NIST and the dieharder*[40] *tests*[41].

(PTG.3.4) The internal random numbers shall *use PTRNG of class PTG.2*.

215 *Application Note 9:* This is a requirement from the Logical Channel package.

35 [selection: *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, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy*]

36 [selection: *externally, at regular intervals, continuously, applied upon specified internal events*]

37 [selection: *bits, octets of bits, numbers* [assignment: *format of the numbers*]]

38 [assignment: *a defined quality metric* ***of the selected RNG class***]

39 [assignment: *additional test suites*]

40 The selected here test suites http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.1.zip and http://www.phy.duke.edu/~rgb/General/dieharder/dieharder-3.31.0.tgz are available at NIST and Dieharder web sites. Note that the dieharder tests include Marsaglia's "Diehard battery of tests" and NIST tests.

41 [assignment: *additional test suites*]

216 *Application Note 10:* The TOE provides random numbers by means of command GET RANDOM for key generation of external devices like the connector (i.e. usage as gSMC-K) or the eHealth card terminals (i.e. usage as SMC-KT). The provided random numbers meet the requirements of TR-03116 [TR3116-1, section 3.5]. Since the command GET RANDOM may be used by the external device to seed another deterministic RNG, the TOE provides this RNG as of class PTG.3 (cf. [AIS31]).

217 FCS_RNG.1/PACE Random number generation – RNG for PACE

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1/PACE The TSF shall provide a *hybrid deterministic*[42] random number generator **RNG class** *DRG.4*[43] **for PACE protocol** that implements[44]

(DRG.4.1) The internal state of the RNG shall *use PTRNG of class PTG.2 as random source*[45].

(DRG.4.2) The RNG provides forward secrecy.

(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.

(DRG.4.4) The RNG provides enhanced forward secrecy *on condition "session closed or aborted"*[46].

(DRG.4.5) The internal state of the RNG is seeded by *a PTRNG of class PTG.2*[47].

FCS_RNG.1.2/PACE The TSF shall provide random ~~numbers~~ *octets of bits*[48] that meet[49]

(DRG.4.6) The RNG generates output for which $k > 2^{34}$ strings[50] of bit length 128 are mutually different with probability $1-\varepsilon$, with $\varepsilon < 2^{-16}$.

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A[51], *the NIST and the dieharder*[52] *tests*[53].

---

42 [selection: ~~*physical*~~, ~~*non-physical true*~~, ~~*deterministic*~~, *hybrid physical*, *hybrid deterministic*]

43 [selection: *DRG.4*, *PTG.3*]

44 [assignment: *list of security capabilities* ***of the selected RNG***]

45 [selection: *use PTRNG of class PTG.2 as random source*, *have* [assignment: *work factor*], *require* [assignment: *guess work*]]

46 [selection: *on demand*, *on condition* [assignment: *condition*], *after* [assignment: *time*]]

47 [selection: *internal entropy source*, *PTRNG of class PTG.2*, *PTRNG of class PTG.3*, [*other selection*]]

48 [selection: *bits*, *octets of bits*, *numbers* [assignment: *format of the numbers*]]

49 [assignment: ~~*a*~~ *defined quality metric* ***of the selected RNG class***]

50 [assignment: *number of strings*]

51 [assignment: *additional test suites*]

52 The selected here test suites http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.1.zip and http://www.phy.duke.edu/~rgb/General/dieharder/dieharder-3.31.0.tgz are available at NIST and Dieharder web sites. Note that the dieharder tests include Marsaglia's "Diehard battery of tests" and NIST tests.

53 [assignment: *additional test suites*]

218 *Application Note 11:* The random nonces for PACE are generated by the DRG.4 generator according to FCS_RNG.1 (see p. 44).

219 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] justified in [PPCOS]: the dependant SFRs are not applicable because this SFR does not use any keys. FCS_CKM.4 Cryptographic key destruction justified in [PPCOS]: the dependant SFRs are not applicable because this SFR does not use any keys.
FCS_COP.1.1/ SHA	The TSF shall perform hashing[54] in accordance with a specified cryptographic algorithm (1) SHA-1, (2) SHA-256, (3) SHA-384, (4) SHA-512 [55] and cryptographic key sizes none[56] that meet the following: TR-03116 [TR3116-1, section 3.2.1], FIPS 180-4 [FIPS180][57].

220 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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
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[58] and specified cryptographic key sizes 192 bit (168 bit effectively)[59] that meet the following: standard ANSI X9.63 [ANSX9.63][60].

54 [assignment: *list of cryptographic operations*]

55 [assignment: *cryptographic algorithm*]

56 [assignment: *cryptographic key sizes*]

57 [assignment: *list of standards*]

58 [assignment: *cryptographic key generation algorithm*]/[selection: Diffie-Hellman-Protocol compliant to [PKCS#3], ECDH compliant to [ECCTR]]

59 [assignment: *cryptographic key sizes*]

60 [assignment: *list of standards*]

221 **FCS_CKM.1/DH.PACE.PICC Cryptographic key generation – DH by PACE**

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_CKM.1.1/ DH.PACE.PICC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECDH compliant to* [ECCTR][61] using the protocol *id-PACE-ECDH-GM-AES-CBC-CMAC-128 with brainpoolP256r1, id-PACE-ECDH-GM-AES-CBC-CMAC-192 with brainpoolP384r1, id-PACE-ECDH-GM-AES-CBC-CMAC-256 with brainpoolP512r1*[62] and specified cryptographic key sizes 256, 384, 512[63] that meet the following: TR-3110 [EACTR], TR-03111 [ECCTR, section 4.3.1][64].

222 *Application Note 12:* The TOE exchanges a shared secret with the external entity during the PACE protocol, see [EACTR]. This protocol is based on the ECDH protocol compliant to TR-03111 [ECCTR] (i.e. the elliptic curve cryptographic algorithm ECKA). The shared secret is used for deriving the AES session keys for message encryption and message authentication according to [EACTR] for the TSF as required by FCS_COP.1/ PACE.PICC.ENC, and FCS_COP.1/PACE.PICC.MAC. FCS_CKM.1/DH.PACE.PICC implicitly contains the requirements for the hashing functions used for key derivation by demanding compliance to TR-03110 [EACTR].

223 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ COS.3TDES	The TSF shall perform decryption and encryption for secure messaging[65] in accordance with a specified cryptographic algorithm 3TDES in CBC mode[66] and cryptographic key sizes 192 bit (168 bit effectively) [67] that meet the following: TR-03116 [TR3116-1], NIST SP800-67 [SP800-67][68].

61 [assignment: *cryptographic key generation algorithm*]/[selection: Diffie-Hellman-Protocol compliant to [PKCS#3], ECDH compliant to [ECCTR]]

62 [selection: *id-PACE-ECDH-GM-AES-CBC-CMAC-128 with brainpoolP256r1, id-PACE-ECDH-GM-AES-CBC-CMAC-192 with brainpoolP384r1, id-PACE-ECDH-GM-AES-CBC-CMAC-256 with brainpoolP512r1*]

63 [assignment: *cryptographic key sizes*]

64 [assignment: *list of standards*]

65 [assignment: *list of cryptographic operations*]

66 [assignment: *cryptographic algorithm*]

67 [assignment: *cryptographic key sizes*]

224 **FCS_COP.1/CB.3TDES Cryptographic operation – CB 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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

FCS_COP.1.1/ CB.3TDES The TSF shall perform[69]

(1) encryption with negotiated key for command PSO ENCIPHER,
(2) decryption with negotiated key for command PSO DECIPHER,
(3) encryption and decryption with card internal key for commands
   a. MUTUAL AUTHENTICATE,
   b. EXTERNAL AUTHENTICATE
(4) encryption with card internal key for command INTERNAL AUTHENTICATE AND
(5) encryption and decryption for trusted channel PSO ENCIPHER and PSO DECIPHER

in accordance with a specified cryptographic algorithm 3TDES in CBC mode[70] and cryptographic key sizes 192 bit (168 bit effectively)[71] that meet the following: TR-03116 [TR3116-1, section 3.3.1], NIST SP800-67 [SP800-67][72].

225 **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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

FCS_COP.1.1/ COS.RMAC The TSF shall perform

(1) computation and verification of cryptographic checksum for commands
   a. MUTUAL AUTHENTICATE,
   b. EXTERNAL AUTHENTICATE,
(2) computation and verification of cryptographic checksum for secure messaging[73]

in accordance with a specified cryptographic algorithm Retail MAC[74] and cryptographic key sizes 192 bit (168 bit effectively)[75] that meet

68 [assignment: *list of standards*]

69 [assignment: *list of cryptographic operations*]

70 [assignment: *cryptographic algorithm*]

71 [assignment: *cryptographic key sizes*]

72 [assignment: *list of standards*]

73 [assignment: *list of cryptographic operations*]

the following: TR-03116 [TR3116-1], COS Specification [EGK-COS][76].

226 *Application Note 13*: The MAC algorithm denoted as “Retail MAC” in this SFR (a notation taken over from [EGK-COS]) is named “Retail MAC 32” in the next SFR FCS_COP.1/CB.RMAC taken over from the PP. The Protection Profile PPCOS uses also the notations “Retail-MAC” and “Retail MAC”. The referred standard [TR3116-1] uses a different notation (“3TDES - Retail CBC MAC”). To avoid confusion with the standardized in [ISO9797] (“Algorithm 3”) and ANS X9.19 (“Optional Procedure 1”) commonly referred as “retail MAC” this ST will use in the text thoroughly the notation “RMAC” for the specified in [TR3116-1] MAC algorithm.

227 *Application Note 14*: Note that according to [TR3116-1] the RMAC algorithm can only be used until end of 2017.

228 FCS_COP.1/CB.RMAC Cryptographic operation – CB 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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/CB.RMAC	The TSF shall perform (1) computation of cryptographic checksum for command INTERNAL AUTHENTICATE, (2) computation and verification of cryptographic checksum for commands a. PSO COMPUTE CRYPTOGRAPHIC CHECKSUM, b. PSO VERIFY CRYPTOGRAPHIC CHECKSUM, (3) computation and verification of cryptographic checksum for trusted channel[77] in accordance with a specified cryptographic algorithm Retail MAC 32[78] and cryptographic key sizes 192 bit (168 bit effectively)[79] that meet the following: TR-03116 [TR3116-1, section 3.2.2], COS Specification [EGK-COS][80].

229 FCS_COP.1/COS.AES Cryptographic operation – COS for AES

Hierarchical to:	No other components.

74 [assignment: *cryptographic algorithm*]

75 [assignment: *cryptographic key sizes*]

76 [assignment: *list of standards*]

77 [assignment: *list of cryptographic operations*]

78 [assignment: *cryptographic algorithm*]

79 [assignment: *cryptographic key sizes*]

80 [assignment: *list of standards*]

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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ COS.AES	The TSF shall perform (1) encryption and decryption with card internal key for command 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) decryption and encryption for secure messaging [81] in accordance with a specified cryptographic algorithm AES in CBC mode[82] and cryptographic key sizes 128 bit, 192 bit, 256 bit[83] that meet the following: TR-03116 [TR3116-1], COS Specification [EGK-COS],FIPS 197 [FIPS197][84].

230 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_CKM.1.1/ AES.SM	The TSF shall generate **session** cryptographic keys in accordance with a specified cryptographic key generation algorithm Key Derivation Function for AES as specified in [ECCTR, sec. 4.4.3][85] and specified cryptographic key sizes 128 bit, 192 bit, 256 bit[86] that meet the following: TR-03111 [ECCTR], COS Specification [EGK-COS], FIPS 197 [FIPS197][87].

231 *Application Note 15*: The Key Generation FCS_CKM.1/AES.SM is used during MUTUAL AUTHENTICATE, EXTERNAL AUTHENTICATE, INTERNAL AUTHENTICATE or GENERAL AUTHENTICATE with establishment of secure messaging (with option Crypto Box also for trusted channel). The algorithm uses the random numbers generated by the TSF as required by FCS_RNG.1 (class DRG.4).

81 [assignment: *list of cryptographic operations*]

82 [assignment: *cryptographic algorithm*]

83 [assignment: *cryptographic key sizes*]

84 [assignment: *list of standards*]

85 [assignment: *cryptographic key generation algorithm*]/[selection: Diffie-Hellman-Protocol compliant to [PKCS#3], ECDH compliant to [ECCTR]]

86 [assignment: *cryptographic key sizes*]

87 [assignment: *list of standards*]

232 FCS_COP.1/CB.AES Cryptographic operation – CB 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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

FCS_COP.1.1/ CB.AES The TSF shall perform[88]

(1) encryption with negotiated key for command PSO ENCIPHER
(2) decryption with negotiated key for command PSO DECIPHER
(3) encryption and decryption for trusted channel
   a. PSO ENCIPHER,
   b. PSO DECIPHER

in accordance with a specified cryptographic algorithm AES in CBC mode[89] and cryptographic key sizes 128 bit, 192 bit, 256 bit[90] that meet the following: TR-03116 [TR3116-1], COS Specification [EGK-COS], FIPS 197 [FIPS197][91].

233 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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

FCS_COP.1.1/ COS.CMAC The TSF shall perform

(1) computation and verification of cryptographic checksum for commands
   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 [92]

in accordance with a specified cryptographic algorithm CMAC[93] and cryptographic key sizes 128 bit, 192 bit, 256 bit[94] that meet the following: TR-03116 [TR3116-1], COS Specification [EGK-COS], NIST SP

88 [assignment: *list of cryptographic operations*]

89 [assignment: *cryptographic algorithm*]

90 [assignment: *cryptographic key sizes*]

91 [assignment: *list of standards*]

92 [assignment: *list of cryptographic operations*]

93 [assignment: *cryptographic algorithm*]

94 [assignment: *cryptographic key sizes*]

800-38B [SP800-38B][95].

234 FCS_COP.1/CB.CMAC Cryptographic operation – CB 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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ CB.CMAC	The TSF shall perform[96] (1) computation of cryptographic checksum for command INTERNAL AUTHENTICATE, (2) computation and verification of cryptographic checksum for trusted channel a. PSO COMPUTE CRYPTOGRAPHIC CHECKSUM b. PSO VERIFY CRYPTOGRAPHIC CHECKSUM in accordance with a specified cryptographic algorithm CMAC[97] and cryptographic key sizes 128 bit, 192 bit, 256 bit[98] that meet the following: TR-03116 [TR3116-1, section 3.2.2], COS Specification [EGK-COS][99].

235 FCS_COP.1/PACE.PICC.ENC Cryptographic operation – PACE secure messaging encryption

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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ PACE.PICC.ENC	The TSF shall perform decryption and encryption for secure messaging[100]in accordance with a specified cryptographic algorithm AES in CBC mode[101] and cryptographic key sizes 128 bit, 192 bit, 256 bit[102] that meet the following: TR-03110 [EACTR, part 2], COS Specification [EGK-COS][103].

95 [assignment: *list of standards*]

96 [assignment: *list of cryptographic operations*]

97 [assignment: *cryptographic algorithm*]

98 [assignment: *cryptographic key sizes*]

99 [assignment: *list of standards*]

100 [assignment: *list of cryptographic operations*]

101 [assignment: *cryptographic algorithm*]

102 [assignment: *cryptographic key sizes*]

103 [assignment: *list of standards*]

236 *Application Note 16*: This SFR requires the TOE to implement the cryptographic primitive for secure messaging with encryption of transmitted data and encrypting the nonce in the first step of PACE. The related session keys are agreed between the TOE and the terminal as part of the PACE protocol according to the FCS_CKM.1/DH.PACE.PICC.

237 FCS_COP.1/PACE.PICC.MAC Cryptographic operation – PACE secure messaging MAC

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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ PACE.PICC.MAC	The TSF shall perform MAC calculation for secure messaging[104] in accordance with a specified cryptographic algorithm CMAC[105] and cryptographic key sizes 128 bit, 192 bit, 256 bit[106] that meet the following: TR-03110 [EACTR, part 2], COS Specification [EGK-COS][107].

238 *Application Note 17*: This SFR requires the TOE to implement the cryptographic primitive for secure messaging with message authentication code over transmitted data. The related session keys are agreed between the TOE and the terminal as part of the PACE protocol according to the FCS_CKM.1/DH.PACE.PICC.

239 FCS_CKM.1/RSA Cryptographic key generation – COS for RSA

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_CKM.1.1/ RSA	The TSF shall generate cryptographic **RSA** keys in accordance with a specified cryptographic key generation algorithm conforming to TR-02102 [TR2102][108] and specified cryptographic key sizes 2048 and 3072 bit modulo length[109] that meet the following: TR-03116 [TR3116-1][110].

104 [assignment: *list of cryptographic operations*]

105 [assignment: *cryptographic algorithm*]

106 [assignment: *cryptographic key sizes*]

107 [assignment: *list of standards*]

108 [assignment: *cryptographic key generation algorithm*]/[selection: Diffie-Hellman-Protocol compliant to [PKCS#3], ECDH compliant to [ECCTR]]

109 [assignment: *cryptographic key sizes*]

110 [assignment: *list of standards*]

240 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_CKM.1.1/ ELC	The TSF shall generate cryptographic **ELC** keys in accordance with a specified cryptographic key generation algorithm *conforming to TR-02102 [TR2102]*[111] **with COS standard curves** and specified cryptographic key sizes 256 bit, 384 bit and 512 bit[112] that meet the following: TR-03111 [ECCTR], COS Specification [EGK-COS][113].

241 *Application Note 18*: The TOE supports only standard elliptic curve parameters listed in the COS Specification [EGK-COS, chap. 6.5]. The parameters implemented in the TCOS are valid for any object file system.

242 *Application Note 19*: The TOE supports the generation of asymmetric key pairs for the following operations:

- qualified electronic signatures,
- authentication of external entities,
- document cipher key decipherment.

243 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ COS.RSA.S	The TSF shall perform (1) PSO COMPUTE DIGITAL SIGNATURE (2) INTERNAL AUTHENTICATE[114] in accordance with a specified cryptographic algorithm (1) RSASSA-PSS-SIGN with SHA-256, (2) RSASSA-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)[115]

111 [assignment: *cryptographic key generation algorithm*]

112 [assignment: *cryptographic key sizes*]

113 [assignment: *list of standards*]

114 [assignment: *list of cryptographic operations*]

115 [assignment: *cryptographic algorithm*]

and cryptographic key sizes 2048 bit and 3072 bit modulo length[116] that meet the following: [TR3116-1], COS Specification [EGK-COS], [PKCS1], [ISO9796-2][117].

244 *Application Note 20*: The TOE supports two variants of the PSO COMPUTE DIGITAL SIGNATURE.

- PSO COMPUTE DIGITAL SIGNATURE without Message Recovery will be used for the signing RSA algorithms RSASSA-PSS-SIGN with SHA-256 (see FCS_COP.1/COS.RSA.S), RSASSA-PKCS1-v1_5 (see FCS_COP.1/COS.RSA.S) and ECDSA with SHA-256, SHA-384 and SHA-512 (see FCS_COP.1/COS.ECDSA.S).
- PSO COMPUTE DIGITAL SIGNATURE with Message Recovery will be used for the for the signing algorithm RSA ISO9796-2 DS2 with SHA-256 (see FCS_COP.1/COS.RSA.S)

245 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
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[118] in accordance with a specified cryptographic algorithm RSA ISO9796-2 DS1[119] and cryptographic key sizes 2048 bit modulo length[120] that meet the following: [TR3116-1], [PKCS1], COS Specification [EGK-COS], [ISO9796-2][121].

246 *Application Note 21*: The command PSO VERIFY CERTIFICATE may store the imported public keys for RSA and ELC temporarily in the *publicKeyList* or permanently in the *persistentCache* or *applicationPublicKeyList*. These keys may be used as authentication reference data for asymmetric key based device authentication (cf. FIA_UAU.5) or user data.

116 [assignment: *cryptographic key sizes*]

117 [assignment: *list of standards*]

118 [assignment: *list of cryptographic operations*]

119 [assignment: *cryptographic algorithm*]

120 [assignment: *cryptographic key sizes*]

121 [assignment: *list of standards*]

247 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ COS.ECDSA.S	The TSF shall perform digital signature generation for commands (1) PSO COMPUTE DIGITAL SIGNATURE (2) INTERNAL AUTHENTICATE[122] in accordance with a specified cryptographic algorithm ECDSA with COS standard curves using (1) SHA-256, (2) SHA-384, (3) SHA-512[123] and cryptographic key sizes 256 bit, 384 bit and 512 bit[124] that meet the following: [TR3116-1], [ECCTR, sec. 4.2.1], COS Specification [EGK-COS], [ANSX9.63][125].

248 **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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ COS.ECDSA.V	The TSF shall perform digital signature verification for import of ELC keys using the commands (1) PSO VERIFY CERTIFICATE (2) PSO VERIFY DIGITAL SIGNATURE (3) EXTERNAL AUTHENTICATE[126] in accordance with a specified cryptographic algorithm ECDSA with COS standard curves using (1) SHA-256, (2) SHA-384, (3) SHA-512[127]

122 [assignment: *list of cryptographic operations*]

123 [assignment: *cryptographic algorithm*]

124 [assignment: *cryptographic key sizes*]

125 [assignment: *list of standards*]

126 [assignment: *list of cryptographic operations*]

127 [assignment: *cryptographic algorithm*]

SHA-256, SHA-384, SHA-512[128] and cryptographic key sizes 256 bit, 384 bit and 512 bit[129] that meet the following: [TR3116-1], [ECCTR], COS Specification [EGK-COS], [ANSX9.63][130].

249 **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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

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 ELC (transcipher of data from ELC to RSA)[131]

in accordance with a specified cryptographic algorithm

(1) for encryption:
   a. RSAES-PKCS1-v1.5 ([RFC3447, 7.2.1]),
   b. RSAES-OAEP ([RFC3447, 7.1.1]),
(2) for decryption:
   a. RSAES-PKCS1-v1.5, ([RFC3447, 7.2.2]),
   b. RSAES-OAEP ([RFC3447, 7.1.2])[132]

and cryptographic key sizes 2048 bit and 3072 bit modulo length for RSA private key operation, 2048 bit length for RSA public key operation, and 256 bit, 384 bit and 512 bit for the COS standard curves[133] that meet the following: [TR3116-1], COS Specification [EGK-COS], [RFC3447][134].

250 **FCS_COP.1/CB.RSA Cryptographic operation – CB RSA**

Hierarchical to: No other components.

128 [assignment: *cryptographic algorithm*]

129 [assignment: *cryptographic key sizes*]

130 [assignment: *list of standards*]

131 [assignment: *list of cryptographic operations*]

132 [assignment: *cryptographic algorithm*]

133 [assignment: *cryptographic key sizes*]

134 [assignment: *list of standards*]

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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

FCS_COP.1.1/ CB.RSA The TSF shall perform encryption with stored key for command PSO ENCIPHER[135]

(1) for encryption:
   a. RSAES-PKCS1-V1.5-ENCRYPT ([RFC3447, 7.2.1]),
   b. RSAES-OAEP-ENCRYPT ([RFC3447, 7.1.1]),
(2) for decryption:
   a. RSAES-PKCS1-V1.5-DECRYPT, ([RFC3447, 7.2.2]),
   b. RSAES-OAEP-DECRYPT ([RFC3447, 7.1.2])[136]

and cryptographic key sizes 2048 bit and 3072 bit modulo length for RSA private key operation, 2048 bit length for RSA public key operation[137] that meet the following:[PKCS1][138].

251 **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] fulfilled

FCS_CKM.4 Cryptographic key destruction fulfilled

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)[139]

in accordance with a specified cryptographic algorithm

(1) for encryption: ELC encryption,
(2) for decryption: ELC decryption[140]

and cryptographic key sizes for RSA keys 2048 and 3072 modulo length and 256 bits, 384 bits, 512 bits for ELC keys with COS stan-

135 [assignment: *list of cryptographic operations*]

136 [assignment: *cryptographic algorithm*]

137 [assignment: *cryptographic key sizes*]

138 [assignment: *list of standards*]

139 [assignment: *list of cryptographic operations*]

140 [assignment: *cryptographic algorithm*]

dard curves[141] that meet the following: [ECCTR], [TR3116-1], [EGK-COS][142].

252 *Application Note 22*: The TOE does not support PSO HASH and ENVELOPE.

253 FCS_COP.1/CB.ELC Cryptographic operation – CB ECC

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] fulfilled FCS_CKM.4 Cryptographic key destruction fulfilled
FCS_COP.1.1/ CB.ELC	The TSF shall perform encryption with stored key for command PSO ENCIPHER[143] in accordance with a specified cryptographic algorithm ELC encryption with COS standard curves and cryptographic key sizes 256 bits, 384 bits, 512 bits[144] that meet the following: [ECCTR, chap. 4.3.1, 4.3.3 and 5.3.1.2][145].

254 *Application Note 23*: The TOE does not support commands PSO HASH and ENVELOPE (cf. [ISO7816]).

255 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] fulfilled
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *physical deletion by overwriting the memory data with zeros, random numbers or the new key*[146] that meets the following: *none*[147].

256 *Application Note 24:* The TOE destroys encryption session keys and the message authentication keys for secure messaging and the PACE protocol after reset or termination of secure messaging session (trusted channel) or reaching fail secure state according to FPT_FLS.1. The TOE clears the memory area of any session keys before starting a new communication with an external entity in a new after-reset-session as required by FDP_RIP.1. A secret key will deleted explicitly after execution of the DELETE command.

141 [assignment: *cryptographic key sizes*]

142 [assignment: *list of standards*]

143 [assignment: *list of cryptographic operations*]

144 [assignment: *cryptographic key sizes*]

145 [assignment: *list of standards*]

146 [assignment: *cryptographic key destruction method*]

147 [assignment: *list of standards*]

257 *Application Note 25:* This SFR covers also the iterated FCS_CKM.4/PACE.PICC from the Contactless Package using the same selections.

6.1.4 Class FIA Identification and Authentication

258 FIA_AFL.1/PIN Authentication failure handling

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication: fulfilled
FIA_AFL.1.1/PIN	The TSF shall detect when ~~an administrator~~ configurable positive integer within 1 to 15[148] unsuccessful authentication attempts occurs related to consecutive failed human user authentication by the PIN via VERIFY, ENABLE VERIFICATION REQUIREMENT, DISABLE VERIFICATION REQUIREMENT or CHANGE REFERENCE DATA commands[149].
FIA_AFL.1.2/PIN	When the defined number of unsuccessful authentication attempts has been met[150], 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[151].

259 *Application Note 26*: 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. authentication attempts is defined in the password objects of the object system. “Consecutive failed authentication attempts” are counted separately for each PIN and interrupted by successful authentication attempt for this PIN, i.e. the PIN object has a *retryCounter* which is initially set to *startRetryCounter*, decremented by each failed authentication attempt and reset to *startRetryCounter* by any successful authentication with the PIN or by 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.

260 FIA_AFL.1/PUC Authentication usage counter

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

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

149 [assignment: *list of authentication events*]

150 [selection: *met, surpassed*]

151 [assignment: *list of actions*]

FIA_AFL.1.1/PUC The TSF shall detect when ~~an administrator~~ configurable positive integer within 1 to 15[152] ~~unsuccessful~~[153] authentication attempts occurs related to usage of a password unblocking code using the RESET RETRY COUNTER command[154].

FIA_AFL.1.2/PUC When the defined number of ~~unsuccessful~~ authentication attempts has been met[155], the TSF shall block the password unblocking code[156].

261 *Application Note 27*: The component FIA_AFL.1/PUC addresses the human user authentication by means of a PUC. The configurable positive integer of usage of password unblocking code is defined in the password objects of the object system.

262 *Application Note 28*: The command RESET RETRY COUNTER can be used to change a password or reset a retry counter. Depending on the object system the usage of the command RESET RETRY COUNTER may be restricted to the ability to reset a retry counter only.

263 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: *pwdIdentifier* in *globalPasswordList* and *pwdIdentifier* 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: *keyIdentifier* of the key,
   d. with secure messaging keys: *keyIdentifier* of the key used for establishing the session key[157]

264 FIA_ATD.1/PACE User attribute definition – PACE protocol

Hierarchical to: No other components.

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

153 Refinement: not only unsuccessful but all attempts are counted here.

154 [assignment: *list of authentication events*]

155 [selection: *met, surpassed*]

156 [assignment: *list of actions*]

157 [assignment: *list of security attributes*]

Dependencies:	No dependencies.
FIA_ATD.1.1/ PACE	The TSF shall maintain the following list of security attributes belonging to individual users: (1) for users defined in FIA_ATD.1 (2) additionally for device: authentication state gained with card SCCO[158].

265 **FIA_UAU.1 Timing of authentication**

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification: fulfilled
FIA_UAU.1.1	The TSF shall allow (1) reading the ATR, (2) *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*[159], (3) commands with access control rule ALWAYS for the current life cycle status and depending on the interface, (4) *none*[160] 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.

266 *Application Note 29*: ATR means Cold ATR and Warm ATR (cf. COS specification [EGK-COS], (N019.900)b).

267 **FIA_UAU.4 Single-use authentication mechanisms**

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

158 [assignment: *list of security attributes*]

159 [selection: *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*]

160 [assignment: *list of TSF-mediated actions*]

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*[161].

268 **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[162],

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the following rules[163]:

(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*[164].

269 **FIA_UAU.6 Re-authenticating**

Hierarchical to: No other components.

161 [assignment: *identified authentication mechanism(s)*]

162 [assignment: *list of multiple authentication mechanisms*]

163 [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

164 [assignment: *additional rules describing how the multiple authentication mechanisms provide authentication*]

Dependencies: No dependencies.

FIA_UAU.6.1 The TSF shall re-authenticate the ~~user~~ **sender of a message**[165] under the conditions

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[166].

270 *Application Note 30:* The entities establishing a secure messaging channel respective a trusted channel authenticate each other and agree on symmetric session keys. The sender of a command authenticates its message by MAC calculation for the command (cf. PSO COMPUTE CRYPTOGRAPHIC CHECKSUM using *SK4TC*, cf. Package Crypto Box) and the receiver of the commands verifies the authentication by MAC verification of commands (using *SK4SM*). The receiver of the commands authenticates its message by MAC calculation (using *SK4SM*) and the sender of a command verifies the authentication by MAC verification of responses (cf. PSO COMPUTE CRYPTOGRAPHIC CHECKSUM using *SK4TC*). If secure messaging is used with encryption then the re-authentication includes the encrypted padding in the plaintext as authentication attempt of the message sender (cf. PSO ENCIPHER for commands) and the receiver (cf. secure messaging for responses) and verification of the correct padding as authentication verification by the message receiver (cf. secure messaging for received commands and PSO DECIPHER for received responses). The specification [EGK-COS] states in section 13.1.2 item (N031.600): “This re-authentication is controlled by the external entity (e.g. the connector in the eHealth environment). If no Secure Messaging is indicated in the class byte CLA (see [ISO7816] Clause 5.3.1) and *SessionkeyContext.flagSessionEnabled* has the value *SK4SM*, then the security status of the key that was involved in the negotiation of the session keys MUST be deleted by means of *clearSessionKeys*(...).” Furthermore item (N031.700) states that the security status of the key that was involved in the negotiation of the session keys MUST be deleted by means of *clearSessionKeys*(...) if the check of the command using CMAC (cf. FCS_COP.1/COS.CMAC) or RMAC[167] fails. The TOE does not execute any command with incorrect message authentication code. The TOE checks each command by secure messaging in encrypt-then-authenticate mode based on a MAC, whether it was sent by the successfully authenticated communication partner. The TOE does not execute any command with incorrect MAC. Therefore, the TOE re-authenticates the communication partner connected, if a secure messaging error occurred, and accepts only those commands received from the initially communication partner.

271 FIA_UAU.6/CB Re-authenticating – Trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

165 Refinement: Identification of the concrete user.

166 [assignment: *list of conditions under which re-authentication is required*]

167 The COS specification uses the identifier “Retail-MAC”, the PP “3TDES Retail CBC MAC”. In fact both are identical to RMAC used in this ST.

FIA_UAU.6.1/CB The TSF shall re-authenticate the ~~user~~ **sender of a message**[168] under the conditions

each message received after establishing the secure messaging by successful authentication by execution of the combination of INTERNAL AUTHENTICATE and EXTERNAL AUTHENTICATE, or MUTUAL AUTHENTICATE or GENERAL AUTHENTICATE commands shall be verified as being sent by the authenticated device using the commands PSO VERIFY CRYPTOGRAPHIC CHECKSUM and PSO DECIPHER[169].

272 **FIA_UAU.1/PACE Timing of authentication – PACE**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification: fulfilled

FIA_UAU.1.1/PACE The TSF shall allow

(1) reading the ATS,
(2) to establish a communication channel,
(3) actions allowed according to FIA_UID.1/PACE and FIA_UAU.1,
(4) *none*[170]

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

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

273 **FIA_UAU.4/PACE.PICC Single-use authentication mechanisms**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

PACE Protocol in PCD role according to TR-03116 [TR3116-1], COS Specification [EGK-COS][171].

274 **FIA_UAU.5/PACE.PICC Multiple authentication mechanisms – PACE/PICC protocol**

Hierarchical to: No other components.

Dependencies: No dependencies.

168 Refinement: Identification of the concrete user.

169 [assignment: *list of conditions under which re-authentication is required*]

170 [assignment: *list of TSF-mediated actions*]

171 [assignment: *identified authentication mechanism(s)*]

FIA_UAU.5.1/ PACE.PICC The TSF shall provide
(1) PACE protocol in PICC role according to [EACTR], [EGK-COS] using commands GENERAL AUTHENTICATE,
(2) secure messaging in MAC-ENC mode using PACE session keys according to [EGK-COS, chapter 13], and [EACTR, part 3] in PICC role[172]
to support user authentication.

FIA_UAU.5.2/ PACE.PICC The TSF shall authenticate any user's claimed identity according to the following rules[173]:
the PACE protocol as PICC is used for authentication of the device using PACE protocol in PCD role and secure messaging in MAC-ENC mode using PACE session keys is used to authenticate its commands[174].

275 **FIA_UAU.6/PACE.PICC Re-authenticating – PACE/PICC protocol**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/ PACE.PICC The TSF shall re-authenticate the user under the conditions after successful run of the PACE protocol as PICC each command received by the TOE shall be verified as being sent by the authenticated PCD[175].

276 *Application Note 31:* The TOE running the PACE protocol as PICC specified in [ICAOSAC] checks each command by secure messaging in encrypt-then-authenticate mode based on CMAC whether it was sent by the successfully authenticated terminal (see FCS_COP.1/PACE.PICC.ENC and FCS_COP.1/PACE.PICC.MAC for further details) and sends all responses using secure messaging after successful PACE authentication. The TOE does not execute any command with incorrect message authentication code. Therefore, the TOE re-authenticates the terminal connected, if a secure messaging error occurred, and accepts only those commands received from the initially authenticated terminal (see FIA_UAU.5/PACE.PICC).

277 **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) *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*[176]

172 [assignment: *list of multiple authentication mechanisms*]

173 [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

174 [assignment: *additional rules describing how the multiple authentication mechanisms provide authentication*]

175 [assignment: *list of conditions under which re-authentication is required*]

(3) commands with access control rule ALWAYS for the current life cycle status and depending on the interface,
(4) *none*[177]

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

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

278 **FIA_UID.1/PACE Timing of identification – PACE**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of authentication: fulfilled

FIA_UID.1.1/PACE The TSF shall allow
(1) reading the ATS,
(2) to establish a communication channel,
(3) *none*[178]

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

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

279 **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,[179]

to prove the identity of the TSF itself[180] to an external entity.

280 **FIA_API.1/CB Authentication Proof of Identity – Trusted channel**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1/CB The TSF shall provide
PSO ENCIPHER and PSO COMPUTE CRYPTOGRAPHIC CHECKSUM *SK4TC* used for trusted channel commands[181]

176 [selection: *GET CHALLENGE, MANAGE CHANNEL, MANAGE SECURITY ENVIRONMENT, SELECT*]

177 [assignment: *list of TSF-mediated actions*]

178 [assignment: *list of TSF-mediated actions*]

179 [assignment: *authentication mechanism*]

180 [assignment: *object, authorized user or role*]

181 [assignment: *authentication mechanism*]

to prove the identity of the TSF itself[182] to an external entity.

281 **FIA_USB.1 User-subject binding**

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition: fulfilled

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: *pwdIdentifier* and Authentication Context *globalPasswordList* and *dfSpecificPasswordList*.
(2) for Human User authenticated with PUC: *pwdIdentifier* of corresponding password,
(3) for Device the Role authenticated by RSA based CVC: the Certificate Holder Authorization (*CHA*) in the CVC
(4) for Device the Role authenticated by ECC based CVC: the Certificate Holder Authorization Template (*CHAT*),
(5) for Device the Role authenticated by symmetric key: *keyIdentifier* and Authentication Context[183].

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*.[184]

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[185]:

(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 pass-

182 [assignment: *object, authorized user or role*]

183 [assignment: *list of user security attributes*]

184 [assignment: *rules for the initial association of attributes*]

185 [assignment: *rules for the changing of attributes*]

word 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:
   a. EXTERNAL AUTHENTICATE with symmetric or public keys,
   b. MUTUAL AUTHENTICATE with symmetric or public keys,
   c. GENERAL AUTHENTICATE with mutual ELC authentication and
   d. 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 a password object or a 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 delected folder shall be set to “not authenticated” and (b) all entire keys in *keyReferenceList* and *allPublicKeyList* related to the delected 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 a 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 delected).

(9) *none* [186].

282 *Application Note 32:* Note the security attributes of the user are defined by the authentication reference data. The user may choose security attributes of the subjects interface in the power on session and *seIdentifier* by execution of command MANAGE SECURITY ENVIRONMENT for the current directory. The initial authentication state is set when the command SELECT is executed and the *newFile* is a folder (cf. COS Specification [EGK-COS], clause (N076.100) and (N048.200)).

186 [assignment: *further rules for the changing of attributes*]

283 **FIA_USB.1/CB User-subject binding – Trusted channel**

Hierarchical to:	No other components.
Dependencies:	FIA_ATD.1 User attribute definition: fulfilled
FIA_USB.1.1/CB	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: as defined in FIA_USB.1[187].
FIA_USB.1.2/CB	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: as defined in FIA_USB.1.[188]
FIA_USB.1.3/CB	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users[189]: (1) If the message received in commands PSO VERIFY CRYPTOGRAPHIC CHECKSUM fails the verification or the message received in command PSO DECIPHER fail the padding condition the authentication state of the user bound to the *SK4TC* is changed to " not authenticated" (i.e. the *keyReferenceList.macCalculation*, *keyReferenceList. dataEncipher* and the *SK4TC* are deleted). (2) *none* [190].

284 **FIA_USB.1/PACE.PICC User-subject binding – PACE/PICC protocol**

Hierarchical to:	No other components.
Dependencies:	FIA_ATD.1 User attribute definition: fulfilled
FIA_USB.1.1/ PACE.PICC	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: The authentication state for the device using PACE protocol in PCD role with a. *keyIdentifier* of the used SCCO in the *globalSecurityList* if SCCO was in MF or in *dfSpecificSecurityList* if the SCCO was in the respective folder, b. *SK4SM* referenced in *Kmac* and *SSCmac*[191].
FIA_USB.1.2/ PACE.PICC	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: see FIA_USB.1[192].

187 [assignment: *list of user security attributes*]

188 [assignment: *rules for the initial association of attributes*]

189 [assignment: *rules for the changing of attributes*]

190 [assignment: *further rules for the changing of attributes*]

191 [assignment: *list of user security attributes*]

192 [assignment: *rules for the initial association of attributes*]

FIA_USB.1.3/ PACE.PICC The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users[193]:

(1) The authentication state for the device after successful authenticated using PACE protocol in PCD role is set to “authenticated” and:
   a. *keyIdentifier* of the used SCCO in the *globalSecurityList* if SCCO was in MF or in *dfSpecificSecurityList* if the SCCO was in the respective DF,
   b. the authentication reference data *SK4SM* is stored in *Kmac* and *SSCmac*.

(2) If an authentication attempt using PACE protocol in PCD role failed
   a. Executing GENERAL AUTHENTICATE for PACE Version 2 [EACTR],
   b. receiving commands failing the MAC verification or encryption defined for secure messaging,
   c. receiving messages violation MAC verification or encryption defined for trusted channel established with PACE

the authentication state for the specific context of SCCO has to be set to “not authenticated” (i.e. the element in *globalSecurityList* respective in the *dfSpecificSecurityList* and the *SK4SM* are deleted).

285 **FIA_USB.1/LC User-subject binding – Logical channel**

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition: fulfilled

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

(1) The authentication state for the context as specified in FIA_USB.1,
(2) The authentication state for a context is bound to the logical channel the authentication took place[194].

FIA_USB.1.2/LC 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 a new logical channel is opened the authentication state is “not authenticated” for all contexts within that logical channel[195].

FIA_USB.1.3/LC The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users[196]:

(1) Every logical channel has its own context. The rules as specified

193 [assignment: *rules for the changing of attributes*]

194 [assignment: *list of user security attributes*]

195 [assignment: *rules for the initial association of attributes*]

196 [assignment: *rules for the changing of attributes*]

in FIA_USB.1.3 for the context shall be enforced for each logical channel separately.
(2) After a logical channel is closed or reset, e.g. by the use of a MANAGE CHANNEL command, the authentication state for all contexts within the closed logical channel must be “not authenticated”
(3) The execution of a DELETE command has to be rejected if more than one channel is open.
(4) *none* [197].

286 **FIA_SOS.1 Specification – Verification**

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*[198].

6.1.5 Class FDP User Data Protection

287 *Application Note 33:* This section defines SFR for access control on User data in the object system. The SFR FDP_ACF.1/MF_DF, FDP_ACF.1/EF, FDP_ACF.1/TEF, FDP_\ ACF.1/SEF and FDP_ACF.1/KEY describe the security attributes of the subject gaining access to these objects. The COS specification [EGK-COS] describes the attributes of logical channels (i.e. subjects in CC terminology) which is valid for the core of COS including all packages. The *globalSecurityList* and *dfSpecificSecurityList* contain all *keyIdentifier* used for successful device authentications, i.e. the list may be empty, may contain a *CHA*, a key identifier of a symmetric authentication key or CAN (in form of the *keyIdentifier* of the derived key) used with PACE.

288 **FDP_ACC.1/MF_DF Subset access control**

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control: fulfilled
FDP_ACC.1.1/ MF_DF	The TSF shall enforce the access control MF_DF_SFP[199] on[200] (1) the subject *logical channel* bind to users a. World, b. Human User, c. Device,

197 [assignment: *further rules for the changing of attributes*]

198 [assignment: *a defined quality metric*]

199 [assignment: *access control SFP*]

200 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

d. Human User and Device,
e. *none*[201],

(2) the objects
a. all executable code implemented by the TOE,
b. MF,
c. DF,
d. Application,
e. Application dedicated file,
f. persistent stored public keys,
g. *none*[202],

(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,
h. *none*[203].

289 *Application Note 34*: Note the commands ACTIVATE, DEACTIVATE and TERMINATE DF for current file applicable to MF, DF, Application and Application dedicated file manage the security life cycle attributes. Therefore access control rules of these commands are described by FMT_MSA.1/Life. The object "all executable code implemented by the TOE" includes IC Dedicated Support Software, the Card Operating System and application specific code loaded on the smartcard by command LOAD CODE or any other means.

290 **FDP_ACF.1/MF_DF Security attribute based access control**

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control: fulfilled FMT_MSA.3 Static attributes initialization: fulfilled
FDP_ACF.1.1/ MF_DF	The TSF shall enforce the access control MF_DF_SFP[204] to objects based on the following[205]: (1) the subject *logical channel* with security attributes a. *interface*, b. *globalPasswordList*, c. *globalSecurityList*, d. *dfSpecificPasswordList*, e. *dfSpecificSecurityList*, f. *bitSecurityList*, g. *SessionkeyContext*,

201 [assignment: *list of further subjects*]

202 [assignment: *list of further objects*]

203 [assignment: *all other operations applicable to MF and DF*]

204 [assignment: *access control SFP*]

205 [assignment: *list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

h. *none*[206]

(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. *none*[207]

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[208]:

(1) SELECT is *ALWAYS allowed*,[209]
(2) GET CHALLENGE is *ALWAYS allowed*,[210]
(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 DF, Application or Application DF 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 DF, Application or Application DF 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

206 [assignment: *further subjects listed in FDP_ACC.1.1/MF_DF with their security attributes*]

207 [assignment: *further subjects listed in FDP_ACC.1.1/MF_DF with their security attributes*]

208 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

209 [selection: *ALWAYS allowed*, [assignment: *supported access control rules*]]

210 [selection: *ALWAYS allowed*, [assignment: *supported access control rules*]]

if it is allowed to execute the command FINGERPRINT in the current folder,
(8) All subjects are allowed to execute command LIST PUBLIC KEY to export all persistent stored public keys[211],
(9) *none*[212].

FDP_ACF.1.3/ MF_DF: The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[213].

FDP_ACF.1.4/ MF_DF: The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[214].

291 *Application Note 35*: The object system defines sets of access control rules depending on the life cycle status, security environment and the interface used (i.e. contact based or contactless interface). The security environment may be chosen for the current folder by means of command MANAGE SECURITY ENVIRONMENT. The command SELECT is therefore pre-requisite for many other commands. The access control rule defines for each command, which is defined by CLA, INS, P1 and P2 and acceptable for the type of the object, the necessary security state, which is reached by successful authentication of human user and devices, to allow the access to the selected object. Note that the command FINGERPRINT process the data representing the TOE implementation like user data (i.e. hash value calculation, no execution or interpretation as code) and is developer specific.

292 *Application Note 36*: The access rules for the execution of the FINGERPRINT command are defined in the object system.

293 FDP_ACC.1/EF Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control: fulfilled

FDP_ACC.1.1/EF The TSF shall enforce the access control EF SFP[215] on[216]
(1) the subject *logical channel* bind to users
a. World,
b. Human User,
c. Device,
d. Human User and Device,
e. *none*[217],
(2) the objects
a. EF,
b. Transparent EF,
c. Structured EF,

211 [assignment: *list of security attributes of subjects*]

212 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

213 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

214 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

215 [assignment: *access control SFP*]

216 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

217 [assignment: *list of further subjects*]

d. *none*[218],

(3) the operation by command following

a. SELECT,

b. DELETE of the current file,

c. *none*[219].

294 *Application Note 37*: Note the commands ACTIVATE, DEACTIVATE and, TERMINATE DF for current file applicable to EF, Transparent EF and Structured EF manage the security life cycle attributes. Therefore access control rules of these commands are described by FMT_MSA.1/Life. The commands CREATE, GET DATA, GET RESPONSE and PUT DATA are optional and not implemented by the TOE. The commands specific for transparent files are described in FDP_ACC.1/TEF and FDP_ACF.1/TEF SFR. The commands specific for structured files are described in FDP_ACC.1/SEF and FDP_ACF.1/SEF SFR.

295 **FDP_ACF.1/EF Security attribute based access control**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control: fulfilled

FMT_MSA.3 Static attributes initialization: fulfilled

FDP_ACF.1.1/EF The TSF shall enforce the access control EF SFP[220] to objects based on the following[221]:

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

a. *interface*,

b. *globalPasswordList*,

c. *globalSecurityList*,

d. *dfSpecificPasswordList*,

e. *dfSpecificSecurityList*,

f. *bitSecurityList*,

g. *SessionkeyContext*,

h. *none* [222]

(2) the objects

a. EF with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the EF and *no transaction protection*[223],

b. *none*[224].

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[225]:

218 [assignment: *list of further objects*]

219 [assignment: *all other operations applicable to MF and DF*]

220 [assignment: *access control SFP*]

221 [assignment: *list of subjects and objects controlled under the indicated SFP, and, for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

222 [assignment: *further subjects listed in FDP_ACC.1.1/EF with their security attributes*]

223 [selection: *transaction protection Mode, checksum*]

224 [assignment: *further subjects listed in FDP_ACC.1.1/EF with their security attributes*]

225 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

(1) SELECT is *ALWAYS allowed*[226],
(2) A subject is allowed to delete the current EF 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*, *interfaceDependentAccessRules* and *seIdentifier* of the current folder.
(3) *none*[227].

FDP_ACF.1.3/EF The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[228].

FDP_ACF.1.4/EF The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[229].

296 **FDP_ACC.1/TEF Subset access control**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control: fulfilled

FDP_ACC.1.1/TEF The TSF shall enforce the access rule TEF SFP[230] on[231]
(1) the subject *logical channel* bind to users
  a. World,
  b. Human User,
  c. Device,
  d. Human User and Device,
  e. *none*[232],
(2) the objects
  a. Transparent EF,
  b. Structured EF,
  c. *none*[233],
(3) the operation by command following
  a. ERASE BINARY,
  b. READ BINARY,
  c. SET LOGICAL EOF
  d. UPDATE BINARY
  e. WRITE
  f. *none*[234].

226 [selection: *ALWAYS allowed*, [assignment: *supported access control rules*]]

227 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

228 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

229 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

230 [assignment: *access control SFP*]

231 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

232 [assignment: *list of further subjects*]

233 [assignment: *list of further objects*]

234 [assignment: *all other operations applicable to MF and DF*]

297 *Application Note 38*: If the checksum of the data to be read by READ BINARY is malicious then the TOE throws a warning on export.

298 FDP_ACF.1/TEF Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control: fulfilled

FMT_MSA.3 Static attributes initialization: fulfilled

FDP_ACF.1.1/ TEF The TSF shall enforce the access rule TEF SFP[235] to objects based on the following[236]:

(1) the subject *logical channel* with security attributes
   a. *interface*,
   b. *globalPasswordList*,
   c. *globalSecurityList*,
   d. *dfSpecificPasswordList*,
   e. *dfSpecificSecurityList*,
   f. *bitSecurityList*,
   g. *SessionkeyContext*,
   h. *none*[237]

(2) the objects
   a. with security attributes *seIdentifier* of the current folder, *lifeCycleStatus* and *interfaceDependentAccessRules* of the current Transparent EF and *no transaction protection*[238],
   b. *none*[239].

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[240]:

(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*[241].

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

235 [assignment: *access control SFP*]

236 [assignment: *list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

237 [assignment: *further subjects listed in FDP_ACC.1.1/TEF with their security attributes*]

238 [selection: *transaction protection Mode, checksum*]

239 [assignment: *further subjects listed in FDP_ACC.1.1/TEF with their security attributes*]

240 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

241 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

on the following additional rules: none[242].

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*[243].

299 *Application Note 39*: The selection of “transaction protection Mode” and “checksum” is empty because they are optional in the COS specification [EGK-COS].

300 FDP_ACC.1/SEF Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control: fulfilled

FDP_ACC.1.1/ SEF The TSF shall enforce the access rule SEF SFP[244] on[245]

(1) the subjects *logical channel* bind to users
   a. World,
   b. Human User,
   c. Device,
   d. Human User and Device,
   e. *none*[246],

(2) the objects
   a. record in Structured EF,
   b. *none*[247],

(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*[248].

301 *Application Note 40*: The command WRITE RECORD is optional and not implemented by the TOE.

302 FDP_ACF.1/SEF Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control: fulfilled

242 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

243 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

244 [assignment: *access control SFP*]

245 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

246 [assignment: *list of further subjects*]

247 [assignment: *list of further objects*]

248 [assignment: *all other operations applicable to MF and DF*]

FMT_MSA.3 Static attributes initialization: fulfilled

FDP_ACF.1.1/ SEF	The TSF shall enforce the access rule SEF SFP[249] to objects based on the following[250]: (1) the subject *logical channel* with security attributes a. *interface*, b. *globalPasswordList*, c. *globalSecurityList*, d. *dfSpecificPasswordList*, e. *dfSpecificSecurityList*, f. *bitSecurityList*, g. *SessionkeyContext*, h. *none*[251] (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*[252]
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[253]: (1) The subject is allowed to execute the command listed in FDP_ACC.1.1/TEF for the record of the current Structured 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 Structured EF, and *lifeCycleStatus* of the record. (2) *none*[254]
FDP_ACF.1.3/ SEF	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[255].
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*[256].

303 *Application Note 41*: Keys can be TSF data or user data. As SFR FDP_ACC.1/KEY and FDP_ACF.1/KEY address protection of user data the keys defined in these SFR as objects are user keys only. Keys used for authentication are TSF data and are therefore not in the scope of these two SFR. Please note that the PSO ENCIPHER, PSO DECIPHER,

249 [assignment: *access control SFP*]

250 [assignment: *list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

251 [assignment: *further subjects listed in FDP_ACC.1.1/SEF with their security attributes*]

252 [assignment: *further subjects listed in FDP_ACC.1.1/SEF with their security attributes*]

253 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

254 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

255 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

256 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

PSO COMPUTE CRYPTOGRAPHIC CHECKSUM, and PSO VERIFY CRYPTOGRAPHIC CHECKSUM are used with the *SK4TC* for trusted channel. If these commands are used in the context trusted channel the key used is TSF data and not user data. Therefore the SFR FDP_ACC.1/KEY and FDP_ACF.1/KEY are not applicable on the commands used for trusted channel. The commands PSO COMPUTE CRYPTOGRAPHIC CHECKSUM and PSO VERIFY CRYPTOGRAPHIC CHECKSUM are required by the package Crypto Box.

304 *Application Note 42*: If the checksum of the record to be read does by READ RECORD not match the TOE will block the output.

305 FDP_ACC.1/KEY Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control: fulfilled
FDP_ACC.1.1/ KEY	The TSF shall enforce the access control key SFP[257] on[258] (1) the subject *logical channel* bind to users a. World, b. Human User, c. Device, d. Human User and Device, e. *none*[259], (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-Hellman key exchange f. *none* [260], (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 COMPUTE CRYPTOGRAPHIC CHECKSUM, h. PSO VERIFY CRYPTOGRAPHIC CHECKSUM, i. PSO ENCIPHER, j. PSO DECIPHER, k. PSO TRANSCIPHER, l. *none*[261].

257 [assignment: *access control SFP*]

258 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

259 [assignment: *list of further subjects listed in FDP_ACC.1.1/KEY*]

260 [assignment: *list of further objects listed in FDP_ACC.1.1/KEY*]

261 [assignment: *further operation*]

306 **FDP_ACF.1/KEY Security attribute based access control**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control: fulfilled

FMT_MSA.3 Static attributes initialization: fulfilled

FDP_ACF.1.1/KEY The TSF shall enforce the access control key SFP[262] to objects based on the following[263]:

(1) the subject *logical channel* with security attributes
   a. *interface*,
   b. *globalPasswordList*,
   c. *globalSecurityList*,
   d. *dfSpecificPasswordList*,
   e. *dfSpecificSecurityList*,
   f. *bitSecurityList*,
   g. *SessionkeyContext*,
   h. *none*[264]

(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*,
   c. 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-Hellman key exchange
   g. *none*[265]

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[266]:

(1) MANAGE SECURITY ENVIRONMENT is *ALWAYS allowed*[267], in cases defined in FDP_ACF.1.4/KEY.

---

262 [assignment: *access control SFP*]

263 [assignment: *list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

264 [assignment: *further subjects listed in FDP_ACC.1.1/KEY with their security attributes*]

265 [assignment: *further subjects listed in FDP_ACC.1.1/KEY with their security attributes*]

266 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

267 [selection: *ALWAYS allowed*, [assignment: *supported access control rules*]]

(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*.

(5) A subject is allowed to compute digital signatures using the private asymmetric key for user data if the security attributes *interface, 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 using 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 *interfaceDependentAccess*Rulest 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*.

(12) *none*[268].

FDP_ACF.1.3/KEY The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[269].

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*[270].

307 **FDP_ACC.1/LC Subset access control**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control: fulfilled

FDP_ACC.1.1/LC The TSF shall enforce the Logical channel SFP[271] on[272]

268 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

269 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

270 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

271 [assignment: *access control SFP*]

(1) the subjects FDP_ACF.1/EF and FDP_ACF.1/MF_DF,
(2) the objects
   a. logical channel
   b. objects as defined in FDP_ACF.1/EF and
   c. objects as defined in FDP_ACF.1/MF_DF,
(3) the operation by command following
   a. command SELECT
   b. command MANAGE CHANNEL to open, reset and close a logical channel[273].

308 **FDP_ACF.1/LC Subset access control – Logical channel**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control: fulfilled

FMT_MSA.3 Static attributes initialization: fulfilled

FDP_ACF.1.1/LC The TSF shall enforce Logical channel SFP[274] to objects based on the following[275]:

(1) the subjects FDP_ACF.1/EF and FDP_ACF.1/MF_DF with security attribute "*logical channel*"
(2) the objects
   a. logical channel with channel number
   b. as defined in FDP_ACF.1/EF and FDP_ACF.1/MF_DF with security attribute "*shareable*"[276].

FDP_ACF.1.2/LC The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed[277]:

(1) The command MANAGE CHANNEL is *ALWAYS allowed*[278].
(2) An subject is allowed to open, reset or close a logical channel with channel number higher than 1 if a logical channel is available and the subject fulfils the access conditions for command MANAGE CHANNEL with the corresponding parameter P1.
(3) An object can be selected as current object in more than one logical channel if it the security attribute "*shareable*" is set to "TRUE"[279].

FDP_ACF.1.3/LC The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[280].

---

272 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

273 [assignment: *all other operations applicable to MF and DF*]

274 [assignment: *access control SFP*]

275 [assignment: *list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

276 [assignment: *further subjects listed in FDP_ACC.1.1/KEY with their security attributes*]

277 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

278 [selection: *ALWAYS allowed,* [assignment: *supported access control rules*]]

279 [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*]

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

if the security attribute of an object is set to "not *shareable*" this object is not accessible as current object in more than one logical channel[281].

309 *Application Note 43*: The COS specification [EGK-COS] claims that the security attribute "*shareable*" is always "TRUE".

310 FDP_IFC.1/SICP Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes: justified by [PP0035, sec. 6.3.2]

FDP_IFC.1.1/SICP The TSF shall enforce the Data Processing Policy[282] on all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software[283].

311 *Application Note 44*: The Data Processing Policy is defined in [PP0035]: User Data and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software.

312 FDP_ITT.1/SICP Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: fulfilled

FDP_ITT.1.1/SICP The TSF shall enforce the Data Processing Policy[284] to prevent the disclosure[285] of user data when it is transmitted between physically-separated parts of the TOE.

313 *Application Note 45*: The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.

280 [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*]

281 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

282 [assignment: *information flow control SFP*]

283 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

284 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

285 [selection: *disclosure, modification, loss of use*]

314 **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 *de-allocation of the resource from*[286] the following objects: password objects, secret cryptographic keys, private cryptographic keys, session keys, *none*[287].

315 **FDP_RIP.1/PACE.PICC Subset residual information protection – PACE/ PICC**

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_RIP.1.1/ PACE.PICC The TSF shall ensure that any previous information content of a resource is made unavailable upon *de-allocation of the resource from*[288] the following objects:

(1) session keys (immediately after closing related communication session),
(2) any ephemeral secret having been generated during DH key exchange
(3) *none*[289].

316 **FDP_SDI.2 Stored data integrity monitoring and action**

Hierarchical to: FDP_SDI.1 Stored data monitoring

Dependencies: No dependencies

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *hardware integrity errors*[290] on all objects, based on the following attributes:

(1) key objects,
(2) PIN objects,
(3) *affectedObject.flagTransactionMode*=TRUE,
(4) *none*[291].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall *enter the hardware security reset state*[292].

286 [selection: *allocation of the resource to, deallocation of the resource from*]

287 [assignment: *other data objects*]

288 [selection: *allocation of the resource to, deallocation of the resource from*]

289 [assignment: *list of additional objects*]

290 [assignment: *integrity errors*]

291 [assignment: *user data attributes*]

317 **FDP_UCT.1/PACE Basic data exchange confidentiality – PACE**

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow]
FDP_UCT.1.1/PACE	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[293] to transmit and receive[294] user data in a manner protected from unauthorized disclosure.

318 **FDP_UIT.1/PACE Data exchange integrity – PACE protocol**

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow] [FDP_ITC.1 Import of user data without security attributes, or FTP_TRP.1 Trusted path]
FDP_UIT.1.1/PACE	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[295] to transmit and receive[296] user data in a manner protected from modification, deletion, insertion, and replay[297].
FDP_UIT.1.2/PACE	The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion, and replay[298] has occurred.

6.1.6 Class FMT Security Management

319 *Application Note 46*: The SFR FMT_SMF.1 and FMT_SMR.1 provide basic requirements to the management of the TSF data.

320 **FMT_SMF.1 Specification of Management Functions**

Hierarchical to:	No other components.
Dependencies:	No dependencies

292 [assignment: *action to be taken*]

293 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

294 [selection: *transmit, receive*]

295 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

296 [selection: *transmit, receive*]

297 [selection: *modification, deletion, insertion, replay*]

298 [selection: *modification, deletion, insertion, replay*]

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions[299]:

(1) Initialization,
(2) Personalization,
(3) Life Cycle Management by means of commands GENERATE ASYMMETRIC KEY PAIR, DELETE, LOAD APPLICATION, TERMINATE, TERMINATE DF, TERMINATE CARD USAGE, *none*[300],
(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,
(7) *none*[301].

321 *Application Note 47*: The protection profile BSI-CC-PP-0035-2007 [PP0035] describes initialization and personalization as management functions. The corresponding COS command used is FORMAT. More details on this command are provided in the Administrator's Guidance [TCOSGD] (cf. also FMT_SMR.1, para. 323 on p. 91). The initialization as a management function corresponds to the Object System Installation of the first part of the Life Cycle Phase 6 (cf. Life cycle phase 6 "Smartcard personalization" on p. 10).

322 *Application Note 48*: LOAD APPLICATION creates new objects together with their TSF data (cf. FMT_MSA.1/Life). In case of folders this includes authentication reference data as passwords and public keys. CREATE is an optional command. It is not supported by the TOE.

323 FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification: fulfilled

FMT_SMR.1.1 The TSF shall maintain the roles[302]

(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,

299 [assignment: *list of management functions to be provided by the TSF*]

300 [assignment: *list of further management functions to be provided by the TSF*]

301 [assignment: *list of further management functions to be provided by the TSF*]

302 [assignment: *the authorized identified roles*]

(5) Device authenticated by means of asymmetric key in the role defined by the Certificate Holder Authorization in the CVC,
(6) *Administrator authenticated for Installation or Personalization*.

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

324 *Application Note 49*: The Administrator authenticates herself by secret data with at least 128 bits of entropy. This data is used in the FORMAT command available only in Life Cycle Phases 5 and 6. The authentication data for the Installation and the Personalization Agent can be selected different. Note that this command is additionally bound to fixed usage counter of 32 which cannot be changed.

325 *Application Note 50*: The protection profile BSI-CC-PP-0035-2007 does not explicitly define role because roles are linked to life cycle of the chip not addressed by SFR. Therefore the current PP defines the role "World" relevant for all parts of the TOE (e.g. physical protection) and roles for COS related SFR.

326 *Application Note 51*: Human users authenticate themselves by identifying the password or Multi-reference password and providing authentication verification data to be matched to the secret of the password object or PUC depending on the command used. The role gained by authorization with a password is defined in the security attributes of the objects and related to the identified commands. The authorization status is valid for the same level and in the level below in the file hierarchy as the password object is stored. The role gained by authentication with a symmetric key is defined in the security attributes of the objects and related to the identified commands. The assignment may assign additional role like the role defined for authentication by means of PACE or "none".

327 FMT_SMR.1/PACE.PICC Security roles – PACE/PICC protocol

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification: fulfilled

FMT_SMR.1.1/ PACE.PICC The TSF shall maintain the roles[303]
(1) the roles defined in FMT_SMR.1,
(2) PACE authenticated terminal,
(3) *none*[304].

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

328 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]: fulfilled

FMT_SMR.1 Security roles: fulfilled

FMT_SMF.1 Specification of Management Functions: fulfilled

303 [assignment: *the authorized identified roles*]

304 [assignment: *additional authorized identified roles*]

FMT_MSA.1.1/ 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[305] to restrict the ability to

(1) create[306] **all** security attributes of the new object DF, Application, Application DF, EF, TEF and SEF[307] to subjects allowed execution of command LOAD APPLICATION for the MF, DF, Application or Application dedicated file where the new object is created[308],

(2) change[306] security attributes of the object MF, DF, Application, Application dedicated file, EF, TEF and SEF[309] by means of command LOAD APPLICATION to *none*[310],

(3) change[306] the security attributes *lifeCycleStatus* to „Operational state (active)“[307] to subjects allowed execution of command ACTIVATE for the selected object[308],

(4) change[306] the security attributes *lifeCycleStatus* to „Operational state (Deactivated)“[307] to subjects allowed execution of command DEACTIVATE for the selected object[308],

(5) change[306] the security attributes *lifeCycleStatus* to „Termination state”[307] to subjects allowed execution of command TERMINATE for the selected EF, the key object or the password object[308],

(6) change[306] the security attributes *lifeCycleStatus* to „Termination state”[307] to subjects allowed execution of command TERMINATE DF for the selected DF, Application or Application DF[308],

(7) change[306] the security attributes *lifeCycleStatus* to „Termination state”[307] to subjects allowed execution of command TERMINATE CARD USAGE[308],

(8) query[306] the security attributes *lifeCycleStatus* by means of command SELECT[307] to *ALWAYS allowed*[311]

(9) delete[306] **all** security attributes of the selected object[307] to *subjects allowed execution of command DELETE for the selected object*[312].

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.

329 *Application Note 52*: The elements of the SFR are repeated as refinements to avoid iterations of the same SFR. The command LOAD APPLICATION allows to create new objects and does not allow an update of existing objects and their security attributes (cf. [EGK-COS, (N039.300)]).

305 [assignment: *access control* SFP(s), *information flow control SFP(s)*]

306 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

307 [assignment: *list of security attributes*]

308 [assignment: *the authorized identified roles*]

309 [assignment: *list of security attributes*]

310 [assignment: *the authorized identified roles*]/[selection: *none, subjects allowed execution of command LOAD APPLICATION for the MF, DF, Application, Application dedicated file where the object is updated*]

311 [selection: *ALWAYS allowed*, [assignment: *supported access control rules*]]

312 [assignment: *list of further security attributes with the authorized identified roles*]

330 **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]: fulfilled FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled
FMT_MSA.1.1/ SEF	The TSF shall enforce the access rule SEF SFP[313] to restrict the ability to (1) change[314] the security attributes *lifeCycleStatus* of the selected record to "Operational state (active)" [315] to subjects allowed to execute the command ACTIVATE RECORD[316], (2) change[314] the security attributes *lifeCycleStatus* of the selected record to "Operational state (Deactivated)"[315] to subjects allowed to execute the command DEACTIVATE RECORD[316], (3) delete[314] **all** security attributes of the selected record[315] to subjects allowed to execute the command DELETE RECORD[316], (4) *none*[317]. 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.

331 *Application Note 53*: The elements of the SFR are repeated to avoid iterations of the same SFR.

332 *Application Note 54*: The access rights can be described in FMT_MSA.1/SEF in more detail. The "authorized identified roles" could therefore be interpreted in a wider scope including the context where the command is allowed to be executed.

333 **FMT_MSA.3 Static attribute initialization**

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes: fulfilled FMT_SMR.1 Security roles: fulfilled
FMT_MSA.3.1	The TSF shall enforce the access control MF_DF SFP, access control EF_SFP, access rule TEF_SFP, access rule SEF_SFP and ac-

313 [assignment: *access control* SFP(s), *information flow control SFP(s)*]

314 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

315 [assignment: *list of security attributes*]

316 [assignment: *the authorized identified roles*]

317 [assignment: *list of further security attributes with the authorized identified roles*]

cess control key Control SFP[318] to provide restrictive[319] 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.**

FMT_MSA.3.2 The TSF shall allow the subjects allowed to execute the command LOAD APPLICATION[320] to specify alternative initial values to override the default values when an object or information is created.

334 *Application Note 55*: The refinements provide rules for setting restrictive security attributes after reset.

335 **FMT_MSA.3/LC Static attribute initialization – Logical channel**

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes: fulfilled

FMT_ SMR.1 Security roles: fulfilled

FMT_MSA.3.1/LC The TSF shall enforce the Logical channel SFP[321] to provide restrictive[322] default values for security attributes that are used to enforce the SFP. **After a logical channel is opened the security attributes of the subject associated with this logical channel are set as follows**

(1) *currentFolder* **is** *root*,
(2) *keyReferenceList, globalSecurityList, globalPasswordList, dfSpecificSecurityList, dfSpecificPasswordList, bitSecurityList* **are empty.**
(3) *SessionkeyContext.flagSessionEnabled* **to** *noSK,*
(4) *seIdentifier* **is #1,**
(5) *currentFile* **is undefined.**

FMT_MSA.3.2/LC The TSF shall allow the subjects allowed to execute the command LOAD APPLICATION[323] to specify alternative initial values to override the default values when an object or information is created.

318 [assignment: *access control SFP, information flow control SFP*]

319 [selection choose one of: *restrictive, permissive*, [assignment: *other property*]]

320 [assignment: *the authorized identified roles*]

321 [assignment: *access control* SFP(s), *information flow control SFP(s)*]

322 [selection, choose one of: *restrictive, permissive*, [assignment: *other property*]]

323 [assignment: *the authorized identified roles*]

336 **FMT_MTD.1/PIN Management of TSF data – PIN**

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled
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)[324] to subjects successful authenticated with the old *secret* of this password object[325], (2) set new *secret* and change *transportStatus* to *regularPassword* of the password objects with *transportStatus* equal to *Leer-PIN*[324] to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01)[325], (3) set new *secret* of the password objects by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,00)[324] to subjects successful authenticated with the PUC of this password object [325], (4) set new *secret* of the password objects by means of command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02)[324] to subjects allowed to execute the command RESET RETRY COUNTER with (CLA,INS,P1)= (00,2C,02)[325].

337 *Application Note 56*: The elements of this SFR are repeated to avoid the iterations of the same SFR.

338 *Application Note 57*: The TOE provides access control to the commands depending on the object system. The refinements repeat the structure of the element in order to avoid iteration of the same SFR. 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.

339 **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]: fulfilled FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled

324 [assignment: *other operations*]

325 [assignment: *the authorized identified roles*]

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[326] to restrict the ability to

(1) reset by means of command VERIFY the security attribute retry counter of password objects[327] to subjects successful authenticated with the *secret* of this password object[328],

(2) reset by means of command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00) the security attribute retry counter of password objects[327] to subjects successful authenticated with the old *secret* of this password object[328],

(3) change by means of command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00) the security attribute *transportStatus* from *Transport-PIN* to *regularPassword*[327] to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,00)[328],

(4) change by means of command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01) the security attribute *transportStatus* from *Leer-PIN* to *regularPassword*[327] to subjects allowed to execute the command CHANGE REFERENCE DATA with (CLA,INS,P1)=(00,24,01)[328],

(5) reset by means of command DISABLE VERIFICATION ENVIRONMENT with (CLA,INS,P1)=(00,26,00) the security attribute retry counter of password objects[327] to subjects successful authenticated with the old *secret* of this password object[328],

(6) reset by means of command ENABLE VERIFICATION REQUIREMENT with (CLA,INS,P1)=(00,28,00) the security attribute retry counter of password objects[327] to subjects successful authenticated with the old *secret* of this password object[328],

(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 objects[327] to subjects successful authenticated with the PUC of this password object[328],

(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[327] to subjects allowed to execute the command RESET RETRY COUNTER with (CLA,INS,P1)=(00,2C,02) or (CLA,INS,P1)=(00,2C,03)[328],

(9) query by means of command GET PIN STATUS the security attributes *flagEnabled*, *retry counter*, *transportStatus*[327] to World[328],

(10) enable[329] the security attribute *flagEnabled* requiring authentication with the selected password[330] to subjects authenticated with *password* and allowed to execute the command ENABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,28,00)[328],

326 [assignment: *access control* SFP(s), *information flow control SFP(s)*]

327 [assignment: *other operations*]

328 [assignment: *the authorized identified roles*]

329 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

330 [assignment: *list of security attributes*]

(11) enable[331] the security attribute *flagEnabled* requiring authentication with the selected password[332] to subjects allowed to execute the command ENABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,28,01)[328],

(12) disable[331] the security attribute *flagEnabled* requiring authentication with the selected password[332] to subjects authenticated with *password* and allowed to execute the command DISABLE VERIFICATION ENVIRONMENT (CLA,INS,P1)=(00,26,00)[328],

(13) disable[331] the security attribute *flagEnabled* requiring authentication with the selected password[332] to subjects allowed to execute the command DISABLE VERIFICATION ENVIRONMENT (CLA,INS,P1)=(00,26,01)[328].

340 *Application Note 58*: The elements of the SFR are repeated to avoid iterations of the same SFR.

341 *Application Note 59*: The command DISABLE VERIFICATION REQUIREMENT can be used to disable the need to perform an authentication via password or Multi-Reference password in a specific context. The command ENABLE VERIFICATION REQUIREMENT can be used to enable the need to perform an authentication. The access rights to execute these commands can be limited to specific contexts. For example: The execution of DISABLE VERIFICATION REQUIREMENT should not be allowed for signing applications. The command DISABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,26,01) allows to disable the verification requirement with the PIN. The command ENABLE VERIFICATION REQUIREMENT (CLA,INS,P1)=(00,28,01) allows anybody to enable the verification requirement with the PIN. 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.

342 *Application Note 60*: The TOE provides access control to the commands depending on the object system.

343 FMT_MTD.1/Auth Management of TSF data – Authentication data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled
FMT_MTD.1.1/ Auth	The TSF shall restrict the ability to (1) import by means of commands LOAD APPLICATION[333] the *root* public keys to roles authorized to execute this command[334], (2) import by means of commands PSO VERIFY CERTIFICATE[333] the *root* public keys to roles authorized to execute this command[334], (3) import by means of commands PSO VERIFY CERTIFICATE[333] the

331 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

332 [assignment: *list of security attributes*]

333 [assignment: *other operations*]

334 [assignment: *the authorized identified roles*]

certificates as device authentication reference data to roles authorized to execute this command[334],

(4) select by means of command MANAGE SECURITY ENVIRONMENT[333] the device authentication reference data to *roles authorized to execute this command*[335] [336].

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.

344 *Application Note 61*: The elements of the SFR are repeated to avoid iterations of the same SFR. If *root* public keys are imported according to clause (2) this public key will be stored in the *applicationPublicKeyList* or the *persistentCache* of the object system.

345 *Application Note 62*: The TOE provides access control to the commands depending on the object system.

346 FMT_MSA.1/Auth Management of security attributes – Authentication data

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: fulfilled FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled
FMT_MSA.1.1/ Auth	The TSF shall enforce the access control key SFP[337] to restrict the ability to query[338] the security attributes access control rights set for the key[339] to meet the access rules of command GET SECURITY STATUS KEY of the object dependent on *lifeCycleStatus*, *seIdentifier* and *interfaceDependentAccessRules*[340].

347 FMT_MTD.1/NE Management of TSF data – No export

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

335 [selection: *World, roles authorized to execute this command*]

336 [assignment: *the authorized identified roles*]

337 [assignment: *access control* SFP(s), *information flow control SFP(s)*]

338 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

339 [assignment: *list of security attributes*]

340 [assignment: *the authorized identified roles*]

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

(1) export TSF data according to FPT_ITE.2[341]
   a. public authentication reference data,
   b. security attributes for objects of the object system,
   c. *none*[342]

   to *successfully authenticated Administrator*[343]

(2) export TSF data according to FPT_ITE.2[344] the *none*[345] to *none*[346]

(3) export[347] 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. *none*[348]

   and the following user data
   i. Private keys of the user,
   j. Symmetric keys of the user,
   k. *none*[349]

   to nobody[350].

348 **FMT_MTD.1/PACE.PICC Management of TSF data – PACE/PICC protocol**

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles: fulfilled FMT_SMF.1 Specification of Management Functions: fulfilled
FMT_MTD.1.1/ PACE.PICC	The TSF shall restrict the ability to read[351] the (1) SCCO used for PACE protocol in PICC role, (2) session keys of secure messaging channel established using PACE protocol in PICC role[352]

341 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

342 [assignment: *list of security attributes*]

343 [assignment: *the authorized identified roles*]

344 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

345 [assignment: list of all TOE specific security attributes not described in COS specification [EGK-COS]]

346 [assignment: *list of types of TSF data*]

347 [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

348 [assignment: *list of types of TSF data*]

349 [assignment: *list of security attributes*]

350 [assignment: *the authorized identified roles*]

351 [assignment: *other operations*]

352 [assignment: *list of TSF data*]

to none[353].

349 *Application Note 63*: The derived session keys *SM4SM* shall be kept secret.

350 FMT_LIM.1/SICP Limited capabilities

Hierarchical to:	No other components.
Dependencies:	FMT_LIM.2 Limited availability: fulfilled by FMT_LIM.2.
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/SICP)' the following policy is enforced: Deploying Test Features after TOE Delivery do not allow TSF data or User Data to be manipulated or disclosed[354].

351 FMT_LIM.2/SICP Limited availability

Hierarchical to:	No other components.
Dependencies:	FMT_LIM.1 Limited capabilities: fulfilled by FMT_LIM.1.
FMT_LIM.2.1/ SICP	The TSF shall be designed 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 do not allow TSF data or User Data to be manipulated or disclosed[355].

6.1.7 Class FPT Protection of the Security Functions

352 The TOE shall prevent inherent and forced illicit information leakage for User Data and TSF-data. The security functional requirement FPT_EMS.1 addresses the inherent leakage. With respect to the forced leakage they have to be considered in combination with the security functional requirements "Failure with preservation of secure state (FPT_FLS.1)" and "TSF testing (FPT_TST.1)" on the one hand and "Resistance to physical attack (FPT_PHP.3)" on the other. The SFRs "Limited capabilities (FMT_LIM.1)", "Limited availability (FMT_LIM.2)" and "Resistance to physical attack (FPT_PHP.3)" together with the SAR "Security architecture description" (ADV_ARC.1) prevent bypassing, deactivation and manipulation of the security features or misuse of TOE functions.

353 FPT_EMS.1 TOE Emanation

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

353 [assignment: *the authorized identified roles*]

354 [assignment: *Limited capability and availability policy*]

355 [assignment: *Limited capability and availability policy*]

FPT_EMS.1.1 The TOE shall not emit *power variations, timing variations during command execution*[356] in excess of *non-useful information*[357] enabling access to following TSF data[358]
(1) Regular password,
(2) Multi-Reference password,
(3) PUC,
(4) Session keys,
(5) Symmetric authentication keys,
(6) Private authentication keys,
(7) *none*[359]
and the following user data[360]
(8) Private asymmetric keys,
(9) Symmetric keys,
(10) *none*[361].

FPT_EMS.1.2 The TSF shall ensure any users[362] are unable to use the following interface smart card circuit contacts[363] to gain access to the following TSF data[364]
(1) Regular password,
(2) Multi-Reference password,
(3) PUC,
(4) Session keys,
(5) Symmetric authentication keys,
(6) Private authentication keys,
(7) *none*[365]
and the following user data[366]
(8) Private asymmetric keys
(9) Symmetric keys
(10) *none*[367].

354 **FPT_EMS.1/PACE.PICC TOE Emanation – PACE/PICC protocol**

Hierarchical to: No other components.

Dependencies: No dependencies.

356 [assignment: *types of emissions*]
357 [assignment: *specified limits*]
358 [assignment: *list of types of TSF data*]
359 [assignment: *list of additional types of TSF data*]
360 [assignment: *list of types of user data*]
361 [assignment: *list of additional types of user data*]
362 [assignment: *type of users*]
363 [assignment: *type of connection*]
364 [assignment: *list of types of (further) TSF data*]
365 [assignment: *list of additional types of TSF data*]
366 [assignment: *list of types of user data*]
367 [assignment: *list of additional types of user data*]

FPT_EMS.1.1/ PACE.PICC — The TOE shall not emit *power variations, timing variations during command execution*[368] in excess of *non-useful information*[369] enabling access to[370]

(1) SCCO,
(2) PACE session keys,
(3) any ephemeral secret having been generated during DH key exchange,
(4) any object listed in FPT_EMS.1
(5) *none*[371]

and *none*[372].

FPT_EMS.1.2/ PACE.PICC — The TSF shall ensure any users[373] are unable to use the following interface the contactless interface and circuit contacts[374] to gain access to[375]

(1) SCCO,
(2) PACE session keys,
(3) any ephemeral secret having been generated during DH key exchange,
(4) any object listed in FPT_EMS.1
(5) *none*[376]

and *none*[377].

355 **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)[378] when shared between the TSF and another trusted IT product..

FPT_TDC.1.2 — The TSF shall use [EGK-COS, chapter 7] “CV-Certificate” and [EGK-COS, Appendix H] “CV-Certificate for ELC-keys”[379] when interpreting the TSF data from another trusted IT product.

368 [assignment: *types of emissions*]
369 [assignment: *specified limits*]
370 [assignment: *list of types of TSF data*]
371 [assignment: *list of additional types of TSF data*]
372 [assignment: *list of types of user data*]
373 [assignment: *type of users*]
374 [assignment: *type of connection*]
375 [assignment: *list of types of (further) TSF data*]
376 [assignment: *list of additional types of TSF data*]
377 [assignment: *list of types of user data*]
378 [assignment: *list of TSF data types*]
379 [assignment: *list of interpretation rules to be applied by the TSF*]

356 **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 [EGK-COS][380].

FPT_ITE.1.2 The TSF shall use[381] *CMAC based fingerprint of the TOE implementation using AES128 with cryptographic key size128 bit that meet the following standard* [SP800-38B][382] for the exported data.

357 *Application Note 64*: The command FINGERPRINT calculates CMAC based fingerprint over the complete executable code actually implemented by the TOE. The TOE implementation includes IC Dedicated Support Software, the Card Operating System and application specific code loaded on the smartcard by command LOAD CODE or similar means. The hash function respective the CMAC based calculation uses the prefix send in the command FINGERPRINT for “fresh” fingerprints over all executable code, i.e. no precomputed values over fixed parts of the code only.

358 **FPT_ITE.2 Export of TSF data**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITE.2.1 The TOE shall export[383]

(1) all public authentication reference data,
(2) all security attributes of the object system and of all objects of the object system for all commands,
(3) *none*[384]

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[385].

FPT_ITE.2.2 The TOE shall use *binary TLV encoding*[386] for the exported data.

380 [assignment: *conditions for export*]

381 [assignment: *list of generation rules to be applied by TSF*]

382 [selection: *SHA-256 based fingerprint of the TOE implementation*, *SHA-384 based fingerprint of the TOE implementation*, *SHA-512 based fingerprint of the TOE implementation*, *CMAC based fingerprint of the TOE implementation using* [selection: *AES128, AES-192, AES-256*] *with cryptographic key size* [selection: *128, 192, 256*] *bit that meet the following standard* [selection:*FIPS180-4, SP800-38B*]/[assignment: *list of generation rules to be applied by the TSF*]

383 [assignment: *list of types of TSF data*]

384 [assignment: *list of all TOE specific security attributes not described in COS specification* [EGK-COS]]

385 [assignment: *conditions for export*]

386 [assignment: *list of encoding rules to be applied by TSF*]

359 *Application Note 65*: The public TSF data addressed as TSF data in bullet (1) in the element FPT_ITE.2.1 covers at least all root 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 (cf. [EGK-COS], *persistentPublicKeyList* in [EGK-COS] and [EGK-WRP], *applicationPublicKeyList* and *PersistentCache* in [EGK-COS]). The bullet (2) in the element FPT_ITE.2.1 covers all security attributes of the object system (cf. [EGK-COS], (N019.900), [EGK-WRP], *objectLocator* (E0) and of all objects with types listed in Table 14 and all TOE specific security attributes and parameters (except secrets). The COS specification [EGK-COS] identifies optional functionality the TOE may support. The TOE (as COS, wrapper and guidance documentation) provides to the user the command GET CARD INFO 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 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 (cf. [EGK-WRP]). The exported data will be encoded by wrapper to allow interpretation of the TSF data. The encoding rules meet the requirements of the Technical Guidance describing the verification tool used for examination of the object system against the specification of the object system ([TR3143]).

360 **FPT_ITE.2/PACE Export of TSF data PACE – protocol**

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITE.2.1/PACE	The TOE shall export[387] (1) the public TSF data as defined in FPT_ITE.2.1 given the following conditions (1) conditions as defined in FPT_ITE.2.1, (2) no export of the SCCO[388].
FPT_ITE.2.2/PACE	The TOE shall use *binary TLV encoding*[389] for the exported data..

361 **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 causing a TOE malfunction, (2) Failure detected by TSF according to FPT_TST.1[390].

387 [assignment: *list of types of TSF data*]

388 [assignment: *conditions for export*]

389 [assignment: *list of encoding rules to be applied by TSF*]

390 [assignment: *list of types of failures in the TSF*]

362 *Application Note 66*: The difference in the assignment for FPT_FLS.1/SICP in the Protection Profile [PP0035] is only editorial.

363 FPT_ITT.1/SICP Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_ITT.1.1/SICP The TSF shall The TSF shall protect TSF data from disclosure[391] when it is transmitted between separate parts of the TOE.

364 *Application Note 67*: 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.

365 This requirement is equivalent to FDP_ITT.1 above but refers to TSF data instead of User Data. It refers to the same Data Processing Policy defined under FDP_IFC.1 above.

366 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[392] to the TSF[393] by responding automatically such that the SFRs are always enforced.

367 *Application Note 68*: The TOE will implement appropriate measures to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TOE can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that the TSP could not be violated at any time. Hence, 'automatic response' means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

368 FPT_TST.1 TSF Testing

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_TST.1.1 The TSF shall run a suite of self tests during initial start-up[394] to demonstrate the correct operation of the TSF[395].

391 [selection: *disclosure, modification, loss of use*]

392 [assignment: *physical tampering scenarios*]

393 [assignment: *list of TSF devices/elements*]

FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of TSF data[396].
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of TSF[397].

6.1.8 Class FRU Resource Utilisation

369 FRU_FLT.2/SICP Fault tolerance

Hierarchical to:	FRU_FLT.2
Dependencies:	FPT_FLS.1 Failure with preservation of secure state: fulfilled
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)[398].

6.1.9 Class FTP Inter-TSF trusted channel

370 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[399] to initiate communication via the trusted channel.
FTP_ITC.1.3/TC	The TSF shall initiate[400] communication via the trusted channel for none[401].

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

395 [selection: [assignment: *parts of TSF*], *the TSF*]

396 [selection: [assignment: *parts of TSF data*], *TSF data*]

397 [selection: [assignment: *parts of TSF*], *TSF*]

398 [assignment: *list of types of failures*]

399 [selection: *the TSF, another trusted IT product*]

400 **Refinement:** The trusted IT product is the terminal. The word "initiate" is changed to "enforce", because the TOE is a passive device that cannot initiate any communication, but can enforce secured communication if required for an object of the object system and the TOE can close the trusted channel after integrity violation of a received command.

401 [assignment: *list of functions for which a trusted channel is required*]

371 *Application Note 69*: The TOE responds only to commands establishing secure messaging channels.

372 **FTP_ITC.1/PACE.PICC Inter-TSF trusted channel – PACE/PICC**

Hierarchical to:	No other components.
Dependencies:	No dependencies
FTP_ITC.1.1/ PACE.PICC	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/ PACE.PICC	The TSF shall permit another trusted IT product[402] to initiate communication via the trusted channel.
FTP_ITC.1.3/ PACE.PICC	The TSF shall ~~**initiate**~~ **enforce**[403] communication via the trusted channel for data exchange between the TOE and the external user if required by access control rule of the object in the object system[404].

373 *Application Note 70*: The trusted IT product is the terminal. The TOE enforces the trusted channel by means of PACE protocol after establishing a communication channel and reading the ATS.

6.2 Security Assurance Requirements for the TOE

374 The assurance requirements for the evaluation of the TOE, its development and operating environment are to choose as the predefined assurance package EAL4 augmented by the following components:

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

375 The Protection Profiles BSI-CC-PP0035 [PP0035] and BSI-CC-PP0082 [PPCOS, chap. 6.2.1] define refinements to the TOE Assurance Requirements which are considered by the TOE Developer under the corresponding assurance packages.

6.3 Security Requirements Rationale

376 A detailed justification required for suitability of the security functional requirements to achieve the security objectives is given in the PP ([PPCOS, chap. 6.3.1]) and is therefore not repeated here.

---

402 [selection: *the TSF, another trusted IT product*]

403 **Refinement:** The trusted IT product is the terminal. The word “initiate” is changed to “enforce”, as the TOE is a passive device that cannot initiate any communication. All communication is initiated by the Terminal, and the TOE enforces the trusted channel.

404 [assignment: *list of functions for which a trusted channel is required*]

6.3.1 Rationale for SFR's Dependencies

377 The following table provides an overview for security functional requirements coverage also giving an evidence for sufficiency and necessity of the SFRs chosen. It uses the Tables 24, 28, 31 and 33 from ([PPCOS, chap. 6.3.1]). Note that the SFRs and objectives related to BSI-CC-PP-0035-2007 ([PP0035]) are not duplicated here.

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging	O.Trustedchannel	O.PACE_CHIP	O.Logicalchannel
PP Basic Requirements												
FDP_RIP.1		x										
FDP_SDI.2	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						
FDP_ACF.1/KEY						x						
FMT_MSA.3						x						
FMT_SMF.1						x						
FMT_MSA.1/Life						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						
FCS_RNG.1							x	x				

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging	O.Trustedchannel	O.PACE_CHIP	O.Logicalchannel
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.RSA.S								x				
FCS_COP.1/COS.CMAC								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					
FTP_ITC.1/TC									x			
Crypto Box package												
FIA_API.1/CB										x		
FIA_UAU6/CB										x		
FIA_USB.1/CB										x		
FCS_COP.1/CB.3TDES								x		x		
FCS_COP.1/CB.RMAC								x		x		
FCS_COP.1/CB.AES								x		x		
FCS_COP.1/CB.CMAC								x		x		
FCS_COP.1/CB.ELC								x				
FCS_COP.1/CB.RSA								x				
Package Contactless												
FCS_CKM.1/DH.PACE.PICC								x			x	
FCS_CKM.4/PACE.PICC								x			x	
FCS_COP.1/PACE.PICC.ENC								x			x	
FCS_COP.1/PACE.PICC.MAC								x			x	
FCS_RNG.1/PACE							x				x	
FDP_RIP.1/PACE.PICC		x									x	
FIA_UAU.1/PACE					x	x					x	
FIA_ATD.1/PACE					x	x					x	
FIA_USB.1/PACE.PICC					x	x					x	
FIA_UAU.4/PACE.PICC					x	x					x	
FIA_UAU.5/PACE.PICC					x						x	
FIA_UAU.6/PACE.PICC					x						x	
FIA_UID.1/PACE					x	x					x	
FPT_EMS.1/PACE.PICC					x	x					x	
FDP_UCT.1/PACE											x	
FDP_UIT.1/PACE											x	
FMT_SMR.1/PACE.PICC					x	x					x	
FMT_MTD.1/PACE.PICC		x			x						x	
FPT_ITE.2/PACE				x							x	
FTP_ITC.1/PACE.PICC					x	x					x	

	O.Integrity	O.Confidentiality	O.Resp-COS	O.TSFDataExport	O.Authentication	O.AccessControl	O.KeyManagement	O.Crypto	O.SecureMessaging	O.Trustedchannel	O.PACE_CHIP	O.Logicalchannel
Package Logical channel												
FCS_RNG.1/GR								x				
FIA_USB.1/LC						x						x
FDP_ACC.1/LC						x						x
FDP_ACF.1/LC						x						x
FMT_MSA.3/LC						x						x

Table 17: SFR coverage

378 The dependency analysis for the security functional requirements given in Tables 25, 29, 32 and 34 of the Protection Profile [PPCOS] shows that the mutual support and internal consistency between all defined functional requirements is satisfied or justified.

6.3.2 Security Assurance Requirements Rationale

379 The assurance package of the Protection Profile was chosen based on the pre-defined assurance package EAL4. This package permits 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 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.

380 The selection of the component ALC_DVS.2 provides a higher assurance of the security of the travel document's development and manufacturing especially for the secure handling of the travel document's material.

381 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. It is required in the Protection Profile BSI-CC-PP-0035-2007 [PP0035] and is therefore included in this ST.

382 The selection of the component AVA_VAN.5 provides a higher assurance of the security by vulnerability analysis to assess the resistance to penetration attacks performed by an attacker possessing a high attack potential.

383 The set of *assurance* components being part of EAL4 fulfils all dependencies a priori.

384 The component ALC_DVS.2 has no dependencies.

385 The component ATE_DPT.2 has the following dependencies: ADV_ARC.1, ADV_TDS.3 and ADV_FUN.1. All of these are met or exceeded in the EAL4 assurance package.

386 The component AVA_VAN.5 has the following dependencies: ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_OPE.1, AGD_PRE.1, and ATE_DPT.1. All of these are met or exceeded in the EAL4 assurance package.

387 Note that the Protection Profiles BSI-CC-PP-0035-2007 [PP0035] and BSI-PP-0082 [PPCOS] refined the Security Assurance Requirements ALC_DEL, ALC_DVS, ALC_CMS, ALC_ CMC, ADV_ARC, ADV_FSP, ATE_COV, AGD_OPE, AVA_VAN, ATE_FUN, and ATE_ IND. They are all considered for the TOE.

7 TOE Summary Specification

388 This section presents an overview of the security functionalities implemented by the TOE and the assurance measures applied to ensure their correct implementation.

389 According to the SFRs the TOE provides the following functionalities

- General protection of User data and TSF data
- Identification and authentication
- Access control
- Cryptographic functions
- Protection of communication
- Accuracy of the TOE security functionality /Self-protection

7.1 General Protection of User Data and TSF Data

390 According to the SFRs FDP_ACC.1 and FDP_ACF.1 and their iterations the access to User Data is restricted by defined rules laid down in the certified object system. The details can be found in the corresponding SFPs. Note that the TOE enforces these access rules, but there is no a priori protection of a said object. The access rights may be provided by certificates. The TOE is able to interpret these certificates accordingly (FPT_TDC.1).

391 The TOE provides an export functionality for non-sensitive but important User data and TSF data. The FINGERPRINT command allows the check of the TSF implementation, the export using the wrapper tool allows to check the access rules of an implemented object system (FPT_ITE.1, FPT_ITE.2, FPT_ITE.2/PACE). The TOE runs self tests during initial start-up to ensure the correct function of the TSF (FPT_TST.1).

392 Residual information of sensitive data in previously used resources will not be available after its usage (FDP_RIP.1, FDP_RIP.1/PACE.PICC). Session keys and message authentication keys will be destroyed after reset or termination of the secure messaging channel (FCS_CKM.4). The TOE hides the correlation of power or timing variations and the command execution accessing sensitive user data as different keys and passwords (FPT_EMS.1, FPT_EMS.1/PACE.PICC). In case of a malfunction, operating errors or integrity check failures (FDP_SDI.2) the TOE enters a secure state (FPT_FLS.1, FPT_\ FLS.1/SICP). This is supported by the functional services of the hardware.

393 The TOE executes self tests (FPT_TST.1) 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.

7.2 Identification and Authentication

394 The protocols for identification and authentication of users and devices is described in the COS Specification [EGK-COS]. The roles assigned after successful authentication are listed in FMT_SMR.1 and FMT_SMR.1/PACE.PICC.

395 The security and the reliability of the identification and authentication are supported by the correct key agreement (FIA_UAU.1, FIA_UAU.4, FIA_UAU.5 and FIA_UAU.6) and the quality of random numbers (FCS_RNG.1). This concerns also the authentication via the contactless interface (FIA_UAU.1/PACE, FIA_UAU.4/PACE.PICC, FIA_UAU.5/PACE.PICC and FIA_UAU.6/PACE.PICC). As the authentication state is left, the session keys cannot be used anymore (FCS_CKM.4).

396 User is authenticated with means of PINs and PUCs, which are bounded by corresponding failure or usage counters (FIA_AFL.1/PIN, FIA_AFL.1/PUC, FIA_SOS.1). Device is authenticated by using a correct key derived from the provided certificate and the authentication context (FIA_USB.1, FIA_USB.1/PACE.PICC and FIA_USB.1/LC).

397 Before a user or device is identified only dedicated commands can be executed. This is supported by FIA_UID.1 and FIA_UID.1/PACE.

398 The TOE maintains security attributes according to FIA_ATD.1 and FIA_ATD.1/PACE beside the identity of user and device.

399 The authentication commands are implemented as required by the COS Specification [EGK-COS](FIA_API.1).

7.3 Access Control

400 The access to User Data is restricted according to the different iterations of the SFRs FDP_ACC.1 and FDP_ACF.1.

401 The access to the TOE security functions and the TSF data is controlled by the functionality of the class FMT (FMT_SMF.1, FMT_MSA.1/Life, FMT_MSA.1/SEF, FMT_MSA.1/PIN, FMT_MSA.1/Auth, FMT_MSA.3, FMT_MSA.3/LC).

402 The management of the authentication data and corresponding security attributes is implemented according [EGK-COS] (FMT_MTD.1/Auth, FMT_MTD.1/PACE.PICC, FMT_\MTD.1/PIN, FIA_SOS.1). The TOE disallows the export of session and authentication keys, passwords and other sensitive user and TSF data specified as such in the object system (FMT_MTD.1/NE). Note that the TOE enforces the access rights of elements of the object system, i.e. data specified as unprotected will be exposed by the TOE. For details refer to the Administrator's Guidance [TCOSGD].

7.4 Cryptographic Functions

403 The TOE provides a hybrid deterministic random number generator of class DRG.4 according to [AIS31] (FCS_RNG.1, FCS_RNG.1/PACE). It is based on a random number generator of class PTG.2 provided by the hardware (FCS_RNG.1/SICP). Note that a generator of class PTG.2 is unpredictable but may have a small bias. The random number returned in the GET RANDOM command is based on this PTG.2 (FCS_RNG.1/GR), but additionally an extra post-processing algorithm is applied, which does not reduce the entropy of the input but removes any bias. The random numbers used in the PACE protocol (FCS_RNG.1/PACE) and by the GET CHALLENGE command are generated by the implemented random number generator of class DRG.4.

404 The TOE implements cryptographic checksum functions, including hash functions used for signature verification and key derivation (FCS_COP.1/SHA) and message authentication codes (MACs) addressed by (FCS_COP.1/COS.RMAC, FCS_COP.1/CB.RMAC, FCS_COP.1/COS.CMAC, FCS_COP.1/CB.CMAC, FCS_COP.1/PACE.PICC.MAC).

405 The TOE provides the symmetric encryption algorithm AES with standardized key lengths of 128, 192 and 256 bits (FCS_COP.1/COS.AES, FCS_COP.1/CB.AES, FCS_\ COP.1/PACE.PICC.ENC, FCS_CKM.1/AES.SM) and due to for interoperability reasons the DES in triple mode (TDES) (FCS_COP.1/COS.3TDES, FCS_COP.1/CB.3TDES, FCS_CKM.1/3TDES_SM).

406 The TOE implements asymmetric crypto algorithms used for encryption/decryption, key agreement and digital signatures based on RSA (FCS_CKM.1/RSA, FCS_COP.1/COS. RSA, FCS_COP.1/CB.RSA, FCS_COP.1/COS.RSA.S, FCS_COP.1/COS.RSA.V) and elliptic curves (FCS_CKM.1/ELC, FCS_CKM.1/DH.PACE.PICC, FCS_COP.1/COS.\ ECDSA.S, FCS_COP.1/COS.ECDSA.V, FCS_COP.1/COS.ELC, FCS_COP.1/CB.ELC). The selection of the curve used for ECC based algorithm might be a security issue. The TOE supports only the curves defined in [ECCTR] and [FIPS186], that are required by [EGK-COS].

407 Cryptographic keys are explicitly deleted by overwriting the memory data with zeros or random numbers, e.g. the new key according to FCS_CKM.4 and FCS_CKM.4/PACE.

7.5 Protection of Communication

408 The secure data exchange in a trusted channel is required by FTP_ITC.1/PACE.PICC and FTP_ITC.1/TC. It is supported by cryptographic operations. The TOE enforces a protected communication over the contactless interface by means of the PACE protocol. It is supported by FDP_UCT.1/PACE and FDP_UIT.1/PACE.

409 The randomness of the parameters of the PACE protocol is guaranteed by the RNG class DRG.4 (FCS_RNG.1/PACE).

410 The strength of algorithms for ensuring confidentiality and integrity is supplied by FCS_COP.1/PACE.PICC.ENC and FCS_COP.1/PACE.PICC.MAC.

7.6 Accuracy of the TOE security functionality /Self-protection

411 The operating system of the TOE protects the security functionality of the TOE as soon as it installed during Installation Phase. The TOE will not emit physical or logical data information on security User Data outside the secure channels controlled by the operating system (FPT_EMS.1). User data (FDP_ITT.1/SICP) and TSF data (FPT_ITT.1/ SICP) are protected by the TOE if processed or transferred within different parts of the TOE according to the TOE Data Processing Policy (FDP_IFC.1/SICP).

412 The TOE will resist physical manipulation and probing (FPT_PHP.3/SICP) and enter a secure state in case a failure occur (FPT_FLS.1, FPT_FLS.1/SICP). This functionality is supported also by the hardware, which was approved in a separate evaluation process.

413 To protect the TOE against malfunction the operating conditions must be in the tolerated ranges which is ensured by FRU_FLT.2/SICP.

414 Dedicated test software is no more available after the TOE is finished (FMT_LIM.1/SICP, FMT_LIM.2/SICP). These functions are disabled for the TOE.

415 During TOE manufacturing the chip hardware provides means to store Initialization Data to identify the hardware. This is supported by FAU_SAS.1/SICP.

7.7 TOE SFR Statements

416 For the sake of completeness the TOE Summary Specification of the previous sections is re-ordered once again. All the TOE SFR statements are listed and it is described how they are fulfilled by the TOE. If appropriate then requirements are handled together to avoid needless text duplication.

417 FDP_RIP.1, FDP_RIP.1/PACE.PICC: Residual information of password objects, cryptographic keys static or ephemeral, session keys are deleted explicitly by overwriting with zeros or random numbers, e.g. the new key after de-allocation of the resource. If the security attributes are reset by the TSF, e.g. after a session is closed, the references to the keys become invalid and additionally the memory data is deleted.

418 FDP_SDI.2: The TSF monitor sensitive user data as PIN and key objects for hardware errors by check sums (error detection codes) and hardware functionality. As soon as an error occur the TOE enters a secure state. This requirement is supported by the Memory Access Control Policy of the hardware and the corresponding SFRs of the TOE's hardware (FDP_ACC.1, FDP_ACF.1) [HWST].

419 FPT_FLS.1, FPT_FLS.1/SICP: If the TOE is exposed to external conditions out of defined ranges or other malfunction occur the TOE enters a secure state. This is supported by TSFs provided by the hardware (cf. [HWST, FPT_FLS.1, FPT_PHP.3, FPT_TST.2). The TOE supports "roll back" and "roll forward" in case of power-off events or data loss in communication. A low system frequency sensor is implemented to prevent the TOE from single stepping. Induced errors will be recognized by the hardware and reset is generated.

420 FPT_EMS.1, FPT_EMS.1/PACE.PICC: Both require that the TOE does not emit any information of sensitive user data and TSF data by emissions and via circuit interfaces. This is supported by the Security Feature "Protection against Snooping" of the hardware (cf. [HWST, SF_PS) and the secure access and processing of sensitive User and TSF data.

421 FPT_TDC.1: Card verifiable certificates (CVC) must be interpreted consistently to assign the intended rights to the corresponding card holders. This is supported by the TOE implementing the corresponding communication protocols which include signature verification and padding and format checking (cf. [EGK-COS, chap. 7]).

422 FPT_ITE.1, FPT_ITE.2, FPT_ITE.2/PACE: The export of dedicated TSF data is necessary to select a communication protocol with a dedicated algorithm. Confidential data is never exported. This is enforced by the TOE's access rules. The FINGERPRINT command provides the capability to verify the correctness of the TSF implementation of the TOE. It uses the approved CMAC mechanism as required by [PPCOS].

423 FPT_TST.1: Self tests during start-up demonstrate the correct operation of the TSF and its protection functions. In addition, the TOE's hardware provides an automated continuous user transparent testing of certain functions.

424 FIA_AFL.1/PIN, FIA_SOS.1: The TOE detects unsuccessful authentication attempts in a row with the PIN and blocks the authentication procedure after a defined number is reached. After a successful authentication the counter is reset to its initial value. The TOE enforces assigned minimal length of the PIN. The maximal length restriction is supported by the TOE. It is not a security but an interoperability requirement. Note that these requirements concern the password objects only. The authentication data used for Administrator's authentication is outside their scope and is therefore not restricted by the

given value of *maximumLength*. According to [TCOSGD] the Administrator's authentication data has an entropy of at least 128 bit.

425 FIA_AFL.1/PUC: The TOE counts authentication attempts with the PUC and blocks the corresponding authentication procedure after a defined number is reached. Note that if the PUC is bound to a usage counter by the object systems the TOE will not reset this counter.

426 FIA_ATD.1, FIA_ATD.1/PACE: The TSF maintain the authentication state gained by dedicated security attributes belonging to individual users and devices. This functionality is supported by the COS and is therefore independent of the installed object system.

427 FIA_UAU.1, FIA_UAU.1/PACE: Dedicated actions are allowed or required before the user is authenticated. Any other action requires authentication. This is laid down in the access rules of object system and will be enforced by the COS.

428 FIA_UAU.4, FIA_UAU.4/PACE.PICC: Authentication data cannot be reused. The TSF require the complete protocol to be executed. Ephemeral keys will be deleted according to FDP_RIP.1.

429 FIA_UAU.5, FIA_UAU.5/PACE.PICC: Dedicated commands as given in these SFRs provide the authentication of users by the TOE. Users are authenticated by password objects (PIN), devices by the different AUTHENTICATION commands. The authentication state is maintained by secure messaging channel. If an authentication error occur the authentication state will be reset. Note that the Administrator's authentication bases on a secure messaging as well. The first FORMAT command must be sent always in a secured channel that is setup by the Manufacturer. This is supported by the COS and cannot be changed by the object system.

430 FIA_UAU.6, FIA_UAU.6/CB, FIA_UAU.6/PACE.PICC: As long as the secure messaging channel is kept, the TOE re-authenticates the message sender. Any command breaking the secure messaging channel, being either not authentic or wrong formatted after decryption will reset the authentication status. The Crypto Box SFR requires that authentication uses the trusted channel.

431 FIA_UID.1, FIA_UID.1/PACE: Dedicated actions are allowed, e.g. reading the ATR, or required before the user is identified. Any other action requires identification. This is laid down in the access rules of object system and is enforced by the COS. Note that the access rules for terminated objects are fixed in the COS and cannot by changed by any object system.

432 FIA_API.1, FIA_API.1/CB: Dedicated commands as given in these SFRs provide the authentication of the TSF and the TOE itself. The Crypto Box SFR requires that authentication uses the trusted channel.

433 FMT_SMR.1, FMT_SMR.1/PACE.PICC: These SFRs describe the roles maintained by the TOE: World (the unauthenticated user), Human User authenticated by a password or PIN, Human User authenticated by a PUC, Administrator authenticated as Manufacturer or Personalization Agent, Device authenticated by means of a symmetric or asymmetric key, PACE authenticated terminal. The roles are bound to corresponding authentication data and a fixed set of access rights defined by the access control rules. Administrator's roles authentication is supported by the COS and cannot be changed by any object system.

434 FIA_USB.1, FIA_USB.1/CB, FIA_USB.1/LC, FIA_USB.1/PACE.PICC: The TOE associates security attributes to authenticated users or devices and enforce said rules for changing them by dedicated commands, e.g. changing the authentication state after a

MANAGE CHANNEL command . The Crypto Box SFR requires that authentication is bound to the trusted channel. This is enforced by the TOE's security functions.

435 FDP_ACC.1/EF, FDP_ACF.1/EF, FDP_ACC.1/SEF, FDP_ACF.1/SEF, FDP_ACC.1/TEF, FDP_ACF.1/TEF, FDP_ACC.1/MF_DF, FDP_ACF.1/MF_DF, FDP_ACC.1/Key, FDP_\ ACF.1/Key: The TOE enforces the corresponding access rules SFP for different objects (Elementary File, Structured EF, Transparent EF, MF/DF, key objects). The access rule enforcement is implemented in the COS and cannot be changed by any object system.

436 FDP_ACC.1/LC, FDP_ACF.1/LC: According to the COS-Specification [EGK-COS] the attribute *shareable* for all objects (if they have any) must always set to "TRUE". Therefore these SFRs are fulfilled automatically.

437 FMT_SMF.1: The TOE provides global management functions like Initialization (Installation), Personalization and Life Cycle Management, and also the management of security attributes, passwords objects and device authentication data by dedicated commands.

438 FMT_MSA.1/Life, FMT_MSA.1/SEF: The TOE enforces the access control policy for the management of life cycle relevant security attributes like *lifeCycleStatus*. The dedicated management functions are specified here. Other management functions are not available.

439 FMT_MSA.3, FMT_MSA.3/LC: Initial default values are set by the COS to restrictive values as listed in these SFRs. This concerns the *currentFolder* set to MF, *currentFile* set to non, the security environment set to the default and reset of the session key context.

440 FMT_MTD.1/PIN, FMT_MSA.1/PIN: PIN/password objects can only be changed by dedicated commands VERIFY, CHANGE REFERENCE DATA and RESET RETRY COUNTER. Usage is restricted to authenticated users only. Note that they can also enable or disable the verification. This is implemented in the COS and cannot be changed. The information on the PIN status is freely accessible. Note that disabling the verification requirement should not be allowed for signature application. But this depends on the object system and can only be enforced by the COS if such an access rule is specified in the object system.

441 FMT_MTD.1/Auth, FMT_MSA.1/Auth: Authentication reference data can only be changed by dedicated commands and are restricted to authenticated users/devices only.

442 FMT_MTD.1/NE: Access conditions laid down in the object system restrict the ability to export sensitive TSF data to dedicated roles, other sensitive User data like private keys are not allowed to be exported at all. The TOE enforces these access rules.

443 FMT_MTD.1/PACE.PICC: Secret session keys and other sensitive data of the PACE protocol including the SCCO can never be read out.

444 FCS_RNG.1, FCS_RNG.1/PACE, FCS_RNG.1/SICP: The TOE provides a hybrid deterministic random number generator of class DRG.4, which is based on a random number generator of class PTG.2 provided by the hardware (FCS_RNG.1/SICP). DRG.4 is the highest level of a deterministic random number generator defined in [AIS31].

445 FCS_RNG.1/GR: The TOE provides a physical random number generator of class PTG.3 with a cryptographic post-processing algorithm of class DRG.3. PTG.3 is the highest level of a physical random number generator defined in [AIS31].

446 FCS_COP.1/SHA: The TOE provides the dedicated hash functions SHA-1. SHA-256, SHA-384 and SHA-512 used by internal functions of the TOE, e.g. for key derivation. Note that the weakened collision resistance of SHA-1 has no impact on the key deriva-

tion, for signature creation SHA-1 is not used. The COS ensures the correctness using different checks during the computation.

447 FCS_COP.1/COS.3TDES, FCS_COP.1/CB.3TDES, FCS_CKM.1/3TDES_SM: The TOE uses the DES in triple Mode (TDES with keying option 1), that supports a key length of 192 bits, for encryption and decryption in CBC mode. This algorithm is used also for secure messaging. The COS ensures the correctness using different checks during the computation in the crypto co-processor.

448 FCS_COP.1/COS.RMAC, FCS_COP.1/CB.RMAC: The TOE provides the non-standard RMAC (Retail MAC) algorithm used in MAC computation and verification. For the naming used in this ST refer to Application Note 13 on p. 51. The COS ensures the correctness using different checks during the computation.

449 FCS_COP.1/COS.AES, FCS_COP.1/CB.AES, FCS_COP.1/PACE.PICC.ENC, FCS_\ CKM.1/AES.SM: The TOE uses the AES with standard key sizes of 128, 192 or 256 bits for encryption and decryption in CBC mode. This algorithm is used also for secure messaging established by the PACE protocol. The COS ensures the correctness using different checks during the computation.

450 FCS_COP.1/COS.CMAC, FCS_COP.1/CB.CMAC, FCS_COP.1/PACE.PICC.MAC: The TOE provides the AES-based standard CMAC algorithm used in MAC computation and verification. This algorithm is used also for secure messaging established by the PACE protocol. The COS ensures the correctness using different checks during the computation.

451 FCS_CKM.1/RSA, FCS_COP.1/COS.RSA, FCS_COP.1/CB.RSA, FCS_COP.1/COS.\ RSA.S, FCS_COP.1/COS.RSA.V: The TOE implements RSA key generation, decryption and digital signature creation with 2048 and 3072 bit key lengths. Public key operations RSA encryption and digital signature verification are supported with 2048 bit key lengths. The COS ensures the correctness using different checks during the computation, e.g. to prevent different fault attacks the output of secret key operations is blocked if the corresponding public operation fails.

452 FCS_CKM.1/ELC, FCS_CKM.1/DH.PACE.PICC, FCS_COP.1/COS.ELC, FCS_COP.1/ CB.ELC, FCS_COP.1/COS.ECDSA.S, FCS_COP.1/COS.ECDSA.V: The TOE implements different cryptographic algorithms based on elliptic curves. The standardized prime curves of 256, 384 and 512 bit key lengths are supported by the TOE. The COS ensures the correctness using different checks during the computation.

453 FCS_CKM.4, FCS_CKM.4/PACE.PICC: Cryptographic keys will be destroyed after deallocation by overwriting with zeros or random data, e.g. the new key.

454 FDP_UCT.1/PACE, FDP_UIT.1/PACE: The TOE implements the PACE protocol, which is proven to be secure. The secure channel set up by the protocol prevents the transmitted data to be disclosed, modified, deleted, inserted or replayed.

455 FTP_ITC.1/TC, FTP_ITC.1/PACE.PICC: The TOE implements the standardized secure messaging protocol based on cryptographic algorithms. It installs a trusted channel that supports confidentiality and integrity of transmitted data. The TOE enforces the protected communication over the contactless interface by means of the proven as secure PACE protocol.

456 FRU_FLT.2/SICP: A malfunction of the hardware may occur if the external operating conditions are not in the specified ranges. This is provided by the security feature “Protection Against Modifying Attacks” of the chip’s hardware (cr. [HWST, SF_PMA]).

457 FPT_FLS.1, FPT_FLS.1/SICP: If the TOE is exposed to the external operating conditions out of range or if a failure, e.g. entropy loss of the random number generator, the TOE enters and preserves a secure state. This is supported by chip's hardware too.

458 FMT_LIM.1/SICP, FMT_LIM.2/SICP: Test software available in manufacturing phase must be not available (limited availability) or not relevant (limited capability) for the TOE.

459 FAU_SAS.1/SICP: During TOE manufacturing the chip hardware provides means to store Initialization Data to identify the hardware.

460 FPT_PHP.3/SICP: Physical probing shall avert the disclosure of assets. This function is provided by the security functions of the hardware.

461 FDP_ITT.1/SICP, FPT_ITT.1/SICP, FDP_IFC.1/SICP: User and TSF data are protected by the TOE if processed or transferred within different parts of the TOE according to the TOE Data Processing Policy. This function is provided by the chip hardware.

7.8 Statement of Compatibility

462 This is the statement of compatibility between this Composite Security Target and the Security Target Chip of the underlying hardware [HWST].

7.8.1 Relevance of Hardware TSFs

463 The TOE is equipped with following Security Features to meet the security functional requirements:

464 Relevant:

- SF_PS Protection against Snooping
- SF_PMA Protection against Modification Attacks
- SF_PLA Protection against Logical Attacks
- SF_CS Cryptographic Support

  Cryptographic support includes TDES/3DES (relevant), AES (relevant), RSA (not relevant), EC (not relevant), SHA-2 (SHA-256 and SHA512 – both not relevant), TRNG (relevant).

465 Not relevant:

466 SF_DPM Device Phase Management

7.8.2 Security Requirements

467 Security Functional Requirements

468 The relevant Security Requirements of the TOE and the hardware can be mapped or are not relevant. They show no conflict between each other.

469 Security Requirements of the TOE related to the Composite ST:

470 The following Security Requirements of the TOE are specific for the Operating System and have no conflicts with the underlying hardware.

PP Basic Requirements

- FDP_RIP.1 no conflict
- FPT_TDC.1 no conflict
- FPT_ITE.1 no conflict
- FPT_ITE.2 no conflict
- FPT_TST.1 no conflict
- FIA_AFL.1/PIN no conflict
- FIA_AFL.1/PUC no conflict
- FIA_ATD.1 no conflict
- FIA_UAU.1 no conflict
- FIA_UAU.4 no conflict
- FIA_UAU.5 no conflict
- FIA_UAU.6 no conflict
- FIA_UID.1 no conflict
- FIA_API.1 no conflict
- FIA_SOS.1 no conflict
- FMT_SMR.1 no conflict
- FIA_USB.1 no conflict
- FDP_ACC.1/MF_DF no conflict
- FDP_ACF.1/MF_DF no conflict
- FDP_ACC.1/EF no conflict
- FDP_ACF.1/EF no conflict
- FDP_ACC.1/TEF no conflict
- FDP_ACF.1/TEF no conflict
- FDP_ACC.1/SEF no conflict
- FDP_ACF.1/SEF no conflict
- FDP_ACC.1/KEY no conflict
- FDP_ACF.1/KEY no conflict
- FDP_SDI.2 no conflict
- FMT_MSA.3 no conflict
- FMT_SMF.1 no conflict
- FMT_MSA.1/Life no conflict
- FMT_MSA.1/SEF no conflict
- FMT_MTD.1/PIN no conflict
- FMT_MSA.1/PIN no conflict
- FMT_MTD.1/Auth no conflict
- FMT_MSA.1/Auth no conflict
- FMT_MTD.1/NE no conflict
- FCS_COP.1/SHA no conflict

- FCS_CKM.1/AES.SM no conflict
- FCS_CKM.1/RSA no conflict
- FCS_CKM.1/ELC no conflict
- FCS_COP.1/COS.RSA.S no conflict
- FCS_COP.1/COS.RSA.V no conflict
- FCS_COP.1/COS.ECDSA.S no conflict
- FCS_COP.1/COS.ECDSA.V no conflict
- FCS_COP.1/COS.RSA no conflict
- FCS_COP.1/COS.ELC no conflict
- FCS_CKM.4 no conflict

Crypto Box package

- FIA_API.1/CB no conflict
- FIA_UAU.6/CB no conflict
- FIA_USB.1/CB no conflict
- FCS_COP.1/CB.ELC no conflict
- FCS_COP.1/CB.RSA no conflict

Package Contactless

- FCS_CKM.1/DH.PACE.PICC no conflict
- FCS_CKM.4/PACE.PICC no conflict
- FIA_UAU.1/PACE no conflict
- FIA_ATD.1/PACE no conflict
- FIA_USB.1/PACE.PICC no conflict
- FIA_UAU.4/PACE.PICC no conflict
- FIA_UAU.5/PACE.PICC no conflict
- FIA_UAU.6/PACE.PICC no conflict
- FIA_UID.1/PACE no conflict
- FDP_RIP.1/PACE.PICC no conflict
- FDP_UCT.1/PACE no conflict
- FDP_UIT.1/PACE no conflict
- FMT_SMR.1/PACE.PICC no conflict
- FMT_MTD.1/PACE.PICC no conflict
- FPT_ITE.2/PACE no conflict
- FTP_ITC.1/PACE.PICC no conflict

Package Logical channel

- FIA_USB.1/LC no conflict
- FDP_ACC.1/LC no conflict
- FDP_ACF.1/LC no conflict
- FMT_MSA.3/LC no conflict

471 Note that some of these requirements, especially all FCS_CKM.1 key generation requirements, requirements FCS_COP.1/RSA, FCS_COP.1/ELC and FCS_COP.1/DH.\ PACE.PICC for cryptographic operations and also the requirements on secure and trusted channel FTP_ ITC.1/TC and FTP_ITC.1/PACE.PICC rely on FCS_RNG.1/SICP requirements of the hardware. This is considered as not conflicting, because the latter is also used by FCS_RNG.1 and FCS_RNG.1/GR of the TOE.

472 The remaining Security Requirements of the TOE can be mapped to Security Requirements of the hardware. They show no conflict between each other.

- FPT_FLS.1 matches FPT_FLS.1 of [HWST]
- FPT_EMS.1, FPT_EMS.1/PACE.PICC are supported by the Security Feature SF_PS of the hardware ([HWST]) and the AVA_VAN.5 evaluation
- FCS_COP.1/COS.AES, FCS_COP.1/COS.CMAC, FCS_COP.1/CB.AES, FCS_COP.1/CB.CMAC, FCS_COP.1/PACE.PICC.ENC, FCS_COP.1/PACE.PICC.MAC match FCS_COP.1/AES of [HWST]
- FCS_COP.1/COS.3TDES, FCS_COP.1/COS.RMAC, FCS_CKM.1/3TDES_SM, FCS_COP.1/CB.3TDES, FCS_COP.1/CB.RMAC match FCS_COP.1/DES of [HWST]
- FCS_RNG.1, FCS_RNG.1/GR, FCS_RNG.1/PACE matches FCS_RNG.1 of [HWST]
- FMT_LIM.1 matches FMT_LIM.1 of [HWST] in the pre-usage phase
- FMT_LIM.2 matches FMT_LIM.2 of [HWST] in the pre-usage phase
- FPT_PHP.3 matches FPT_PHP.3 of [HWST]

473 Security Requirements of the hardware

474 The Security Requirements of the TOE's hardware based on PP-0035 [PP0035, sec.6.1] can be mapped to Security Requirements of the TOE. They show no conflict between each other and are taken over in the Composite ST as iterated by SICP.

- FAU_SAS.1 is covered by FAU_SAS.1 of the Composite ST
- FDP_IFC.1 concerns information flow policy between parts of the hardware
- FDP_ITT.1 concerns basic internal transfer protection of the hardware
- FMT_LIM.1 is covered by FMT_LIM.1 of the Composite ST
- FMT_LIM.2 is covered by FMT_LIM.1 of the Composite ST
- FPT_FLS.1 covered by FPT_FLS.1 of the Composite ST
- FPT_ITT.1 concerns basic hardware internal TSF data transfer protection
- FPT_PHP.3 concerns the resistance to physical attacks
- FRU_FLT.2 concerns the hardware operation, does not conflict with SFRs of the TOE

475 The additional Security Requirements of the TOE's hardware defined in [HWST] can be mapped to Security Requirements of the TOE too. They show no conflict between each other.

- FCS_CKM.1 not relevant, as the EC key generation of the hardware is not used
- FCS_COP.1/AES: covered by FCS_COP.1/COS.AES, FCS_COP.1/CB.AES, FCS_COP.1/COS.CMAC and FCS_COP.1/CB.CMAC of the Composite ST
- FCS_COP.1/DES: FCS_COP.1/COS.3TDES, FCS_COP.1/CB.3TDES, FCS_COP.1/COS.RMAC and FCS_COP.1/CB.RMAC of of the Composite ST
- FCS_COP.1/RSA, FCS_COP.1/ECDSA, FCS_COP.1/ECDH, FCS_COP.1/SHA are not relevant, as these algorithms are not used

- FCS_RNG.1: matches FCS_RNG.1 of the Composite ST
- FDP_ACC.1 concerns the Memory Access Control Policy on software tasks accessing assigned data in memories, this is covered by FDP_ACC.1 and its iterations of the Composite TOE
- FDP_ACF.1 describes the Memory Access Control policy enforced by the hardware, this is covered by policy enforcing FDP_ACF.1 of the Composite TOE and its iterations
- FDP_SDI.1, FDP_SDI.2 concern the low-level stored data integrity of the hardware and does not conflict with the SFRs of the TOE.
- FMT_MSA.1 concerns the management of security attributes on hardware's level, does not conflict with the SFRs of the TOE
- FMT_MSA.3 concerns the management of security attributes on hardware's level, does not conflict with the SFRs of the TOE
- FMT_SMF.1 concerns the access of the configuration registers of the Memory Management Unit, does not conflict with the SFRs of the TOE
- FPT_TST.2: concerns self tests of the hardware TSF, no conflicts to SFRs of the TOE

476 **Security Assurance Requirements**

477 The level of assurance of the TOE is EAL 4 augmented with ALC_DVS.2, ATE_DPT.2 and AVA_VAN.5.

478 The chosen level of assurance of the hardware is EAL 5 augmented with ALC_DVS.2 and AVA_VAN.5.

479 This shows that the Assurance Requirements of the TOE matches the Assurance Requirements of the hardware.

7.8.3 Security Objectives

480 The Security Objectives of the TOE and the hardware can be mapped or are not relevant. They show no conflict between each other.

481 **Security Objectives of the TOE related to the Composite ST:**

- O.Integrity: covers O.Add_Functions (AES) and (TDES) of the [HWST]
- O.Confidentiality: covers O.Add_Functions (AES) and (TDES) of the [HWST]
- O.Resp-COS: no conflict
- O.TSFDataExport: no conflict
- O.Authentication: no conflict
- O.AccessControl: no conflict
- O.KeyManagement: no conflict
- O.Crypto: no conflict
- O.SecureMessaging: no conflict

- O.Trustedchannel: no conflict
- O.PACE_CHIP: no conflict
- O.Logicalchannel: no conflict

482 **Security Objectives for the hardware ([PP0035] and [HWST]):**

- O.Identification: is taken over in this ST
- O.Leak-Inherent: is taken over in this ST
- O.Phys-Probing: is taken over in this ST
- O.Malfunction: is taken over in this ST
- O.Phys-Manipulation: is taken over in this ST
- O.Leak-Forced: is taken over in this ST
- O.Abuse-Func: is taken over in this ST
- O.RND: is taken over in this ST
- O.Add-Functions (Additional Specific Security Functionality)

  The hardware TOE provides the following specific security functionality to the Smartcard Embedded Software: Advanced Encryption Standard (AES)/(DES) which is mapped to O.Integrity and O.Confidentiality. The security functionality of Rivest-Shamir-Adleman algorithm, Elliptic Curve Cryptography and Secure Hash Algorithm is not used and therefore not relevant.
- O.Mem_Access

  The hardware TOE provides the Smartcard Embedded Software with the capability to define restricted access memory areas. The hardware TOE enforces the partitioning of such memory areas so that access of software to memory areas and privilege levels is controlled as required. This objective addresses a low-level access control, which does not contradict the access control rules on OS level. The TOE rely on the low-level protection of memory areas and therefore this objective of the hardware is covered by O.Integrity, O.Confidentiality, O.Resp-COS, O.AccessControl.for example, in a multi-application environment. is mapped to T.Mem_Access

7.8.4 Compatibility: TOE Security Environment

483 **Assumptions**

484 The following list shows that assumptions neither of the TOE nor of the hardware have any conflicts between each other. They are either not relevant for this Security Target or are covered by appropriate Security Objectives.

485 **Assumptions for the TOE related to the Composite ST:**

- A.Process-Sec-SC
- A.Plat-COS
- A.Resp-ObjS

486 Assumptions of the Hardware PP ([PP0035]):

- A.Process-Sec-IC (Protection during Packaging, Finishing and Personalization) is coveres by A.Process-Sec-SC
- A.Plat-Appl (Usage of Hardware Platform) not relevant
- A.Resp-Appl (Treatment of User Data) relevant
  This assumption is covered by the hardware's objective for the environment OE.Resp-ObjS

487 Assumptions of the specific hardware platform ([HWST]):

- A.Key-Function (Usage of Key-dependent Functions)
  Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced). This assumption is covered by the Hardware's objective OE.Resp-Appl for the environment and applies to Life Cycle Phase 1 "Development".

488 Threats

489 The Threats of the TOE and the hardware can be mapped or are not relevant. They show no conflict between each other.

490 Threats for the TOE related to the Composite ST:

- T.Forge_Internal_Data no conflict
- T.Compromise_Internal_Data no conflict
- T.Misuse no conflict
- T.Malicious_Application no conflict
- T.Crypto no conflict
- T.Intercept no conflict
- T.WrongRights: no conflict

491 Threats of the hardware ST related to PP0035:

- T.Leak-Inherent is taken over in this ST
- T.Phys-Probing is taken over in this ST
- T.Malfunction is taken over in this ST
- T.Phys-Manipulation is taken over in this ST
- T.Leak-Forced is taken over in this ST
- T.Abuse-Func is taken over in this ST
- T.RND is taken over in this ST

492 **Threats of the hardware ST ([HWST]):**

- T.Mem-Access (Memory Access Violation)

493 Parts of the Smartcard Embedded Software may accidentally or deliberately access restricted data (which may include code) or privilege levels. Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard Embedded Software. This threat is mainly related to TOE's Life Cycle Phase 1 "Development". It is not related to later phases because the Smart Card Embedded Software cannot be altered by the object system.

7.8.5 Organizational Security Policies

494 The Organizational Security Policies of the TOE and the hardware have no conflicts between each other. They are shown in the following list.

495 **Organizational Security Policies of the Composite ST of the TOE:**

- P.Process-TOE covers P.Process-TOE of the hardware ST ([PP0035])
- OSP.Logicalchannel no conflict

496 **Organizational Security Policies of the Hardware ST:**

- P.Add-Functions (Additional Specific Security Functionality) no conflict
  The TOE's hardware provides the following specific security functionality to the Smartcard Embedded Software: Advanced Encryption Standard, Triple Data Encryption Standard (relevant), Rivest-Shamir-Adleman Cryptography (not relevant), Elliptic Curve Cryptography (not relevant), Secure Hash Algorithm SHA-2.
- P.Process-TOE ([PP0035]) is taken over in this ST.

7.8.6 Conclusion

497 No contradictions between the Security Targets of the TOE and the underlying hardware can be found.

7.9 Assurance Measures

498 The documentation is produced compliant to the Common Criteria Version 3.1. The following documents provide the necessary information to fulfill the assurance requirements listed in section 6.2 Security Assurance Requirements for the TOE.

Development

ADV_ARC.1	Security Architecture Description TCOS FlexCert 2.0 Release 1
ADV_FSP.4	Functional Specification TCOS FlexCert 2.0 Release 1
ADV_IMP.1	Implementation of the TSF TCOS FlexCert 2.0 Release 1
ADV_TDS.3	Modular Design of TCOS FlexCert 2.0 Release 1

Guidance documents

AGD_OPE.1 User Guidance TCOS FlexCert 2.0 Release 1
AGD_PRE.1 Administrator Guidance TCOS FlexCert 2.0 Release 1

Life-cycle support
ALC_CMC.4, ALC_CMS.4 Documentation for Configuration Management
ALC_DEL.1 Documentation for Delivery and Operation
ALC_LCD.1 Life Cycle Model Documentation TCOS FlexCert 2.0 Release 1
ALC_TAT.1, ALC_DVS.2 Development Tools and Development Security for TCOS FlexCert 2.0 Release 1

Tests
ATE_COV.2, ATE_DPT.2 Test Documentation for TCOS FlexCert 2.0 Release 1
ATE_FUN.1 Test Documentation of the Functional Testing

Vulnerability assessment
AVA_VAN.5 Independent Vulnerability Analysis TCOS FlexCert 2.0 Release 1

499 The developer team uses a configuration management system that supports the generation of the TOE. The configuration management system is well documented and identifies all different configuration items. The configuration management tracks the implementation representation, design documentation, test documentation, user documentation, administrator documentation, and security flaws. The security of the configuration management is described in detail in a separate document.

500 The delivery process of the TOE is well defined and follows strict procedures. Several measures prevent the modification of the TOE based on the developer's master copy and the user's version. The Administrator and the User are provided with necessary documentation for installation, personalization and start-up of the TOE.

501 The implementation is based on an informal high-level and low-level design of the components of the TOE. The description is sufficient to generate the TOE without other design requirements.

502 The tools used in the development environment are appropriate to protect the confidentiality and integrity of the TOE design and implementation. The development is controlled by a life-cycle model of the TOE. The development tools are well-defined and use semi-formal methods, i.e. a security model.

503 The development department is equipped with organizational and personnel means that are necessary to develop the TOE. The testing and the vulnerability analysis require technical and theoretical know-how available at T-Systems International GmbH.

504 As the evaluation is identified as a composite evaluation based on the CC evaluation of the hardware, the assurance measures related to the hardware (IC) will be provided by documents of the IC manufacturer.

Appendix Glossary and Acronyms

505 The terminology and abbreviations of Common Criteria version 3.1 [CC], Revision 4 and the specification [EGK-COS] apply. The following table is taken over from the PP [PPCOS]

Acronyms

Acronym	Term
CAP	Composed Assurance Package
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
CM	Configuration Management
COS	Card operating system
CVC	Card verifiable certificate
EAL	Evaluation Assurance Level
eHC	Electronic health care card (elektronische Gesundheitskarte)
eHPC	Electronic professional card (elektronischer Heilberufsausweis)
IC	Integrated Circuit
OS	Operating System
OSP	Organizational Security Policy
PC	Personal Computer
PCD	Proximity Coupling Device (as defined in [EACTR part 2])
PICC	Proximity Integrated Circuit Chip (as defined in [EACTR, part 2])
PKI	Public Key Infrastructure
PP	Protection Profile
SAR	Security Assurance Requirement
SCCO	Symmetric Card Connection Object
SFP	Security Function Policy
SFR	Security Functional Requirement
SMC-B	Secure module card type B
SMC-K	Secure module card type K
SMC-KT	Secure module card type KT
SPD	Security Problem Definition
ST	Security Target
TOE	Target of Evaluation

References

[AIS31]
Bundesamt für Sicherheit in der Informationstechnik, Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 31, A proposal for Functionality classes for random number generators Version 2.0 vom 18.09.2011, Bundesamt für Sicherheit in der Informationstechnik (BSI)

[AIS36]
Bundesamt für Sicherheit in der Informationstechnik, Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 36, Version 2 vom 12.11.2007, Bundesamt für Sicherheit in der Informationstechnik (BSI)

[ANSX9.63]
American National Standard X9.63-2001, Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography, 2005-11

[CC]
Common Criteria for Information Technology Security Evaluation, Version 3.1,
Part 1: Introduction and general model; Version 3.1, Sept. 2012, CCMB-2012-09-001,
Part 2: Security functional components; Version 3.1, Sept. 2012, CCMB-2012-09-002,
Part 3: Security assurance components; Version 3.1, Sept. 2012, CCMB-2012-09-003

Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, September 2012, CCMB-2012-09-004

[EACTR]
Technical Guideline TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents,
Part 1 – eMRTDs with BAC/PACEv2 and EACv1,
Part 2 – Extended Access Control Version 2 (EACv2), Password Authenticated Connection Establishment (PACE),and Restricted Identification (RI),
Part 3 – Common Specifications, Version 2.10, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2012-03

[ECCTR]
Technical Guideline TR-03111: Elliptic Curve Cryptography, Version 2.0, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2012-08

[EGK-COS]
Einführung der Gesundheitskarte, Spezifikation des Card Operating System (COS), Elektrische Schnittstelle, Version 3.7.0 vom 26.08.2014, gematik Gesellschaft für Telematikanwendungen der Gesundheitskarte GmbH

Please take also into account the following normative errata published by the gematik:

Errata zu Release 1.4.4, Korrektur der Stapelsignaturfunktion der gSMC-K, Absicherung der kontaktlosen Schnittstelle der eGK, optionale Korrektur der asynchronen symmetrischen Kartenadministration der eGK, Version 1.0.0 vom 07.05.2015

Errata zu Release 1.4.2, Störungsampel, Zertifikate, Testkarten und COS-Wrapper, Version 1.0.1 vom 08.12.2014

2. Errata zu Release 1.4.0, Spezifikation des Card Operating System und Spezifikation Wrapper, Version 1.0.0 vom 06.10.2014

Errata zu Release 1.4.0, Kartenspezifikationen und Konnektor, Version 1.0.0 vom 02.10.2014

[EGK-WRP]
Einführung der Gesundheitskarte, Spezifikation Wrapper, Version 1.6.0 vom 26.08.2014, gematik Gesellschaft für Telematikanwendungen der Gesundheitskarte GmbH

[FIPS180]
Federal Information Processing Standards Publication FIPS PUB 180-4, Secure Hash Standard (SHS), 2012-03

[FIPS186]
Federal Information Processing Standards Publication FIPS PUB 186-4, Digital Signature Standard (DSS), July 2013

[FIPS197]
Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), U.S. Department of Commerce/National Institute of Standards and Technology, 2001-11-26

[HWCR]
Certification Report of the underlying hardware platform, BSI-DSZ-CC-0829-2012 for Infineon Technologies Smart Card IC (Security Controller) M7820 A11and M11, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2012-09

[HWST]
Security Target of the underlying hardware platform, Security Target M7820 A11 and M11, Version 1.6, Infineon Technologies AG, Chipcard and Security, Evaluation Documentation, 2012-08-28

[ICAOSAC]
ICAO Machine Readable Travel Documents, Technical Report, Supplemental Access Control for Machine Readable Travel Documents, Version 1.01, ICAO, 2010-11

[ISO7816]
ISO 7816-4:2013, Identification cards – Integrated circuit cards with contacts, Part 4: Organization, security and commands for interchange, ISO, 2013-04

[ISO9796-2]
ISO/IEC 9796-2:2010 Information technology -- Security techniques -- Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms, ISO, 2010-12

[ISO9797]
ISO 9797-1:1999, Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, ISO, 2005-01-04

[ISO14443]
ISO 14443, Identification cards – Contactless integrated circuit cards – Proximity cards, Parts 1-4 and Amendments, 2008-2014

[PKCS1]
PKCS #1: RSA Cryptography Standard, RSA Laboratories, Version 2.1, Revised June 13, 2002 (cf. [RFC3447])

[PP0035]
Smartcard IC Platform Protection Profile, Version 1.0, 15.06.2007, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik under BSI-CC-PP-0035-2007

[PPCOS]
CC Protection Profile: Card Operating System Generation 2 (PP COS G2), BSI-CC-PP-0082-V2, Version 1.9, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik under BSI-CC-PP-0082-V2-2014, 2014-11

[RFC3447]
J. Jonsson, B. Kaliski; Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447, IETF, 2003-02

[RFC5639]
M. Lochter, J. Merkle, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, RFC 5639, IETF, 2010-03

[SP800-38B]
Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38B, National Institute of Standards and Technology, May 2005

[SP800-67]
Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67, Revised January 2012, National Institute of Standards and Technology, 2012-01

[TCOSGD]
Administrator's Guidance TCOS FlexCert Version 2.0 Release 1, T-Systems International GmbH, Version 1.0, 2015-06
Guidance Documentation of the Wrapper to TCOS FlexCert Version 2.0 Release 1, T-Systems International GmbH, Version 1.0, 2015-06

[TR2102]
Technische Richtlinie TR-02102 Kryptographische Verfahren Empfehlungen und Schlüssellängen, Version 2015-01, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2015-02

[TR3116-1]
Technische Richtlinie TR-03116 für die eCard-Projekte der Bundesregierung Version 3.18, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2014-01

[TR3143]
Technische Richtlinie TR-03143 „eHealth G2-COS Konsistenz-Prüftool", Bundesamt für Sicherheit in der Informationstechnik (BSI), 2015-05