ASE - Security Target

Java Card Platform Embedded Software V3 (Core)

GemXplore'Xpresso V3

	Name	Role	Date (dd/mm/yy)	Visa
Issued by	C. Teri C. Aillaud	CC responsible Project leader	19/02/02 04/04/01	
Verified by	C. Teri O. Marchand	CC responsible Project leader	19/02/02 19/02/02	
Approved by	C. Aillaud O. Marchand	Project leader Project leader	19/02/02 19/02/02	

Authors: Corinne TERI
Christophe AILLAUD

DISTRIBUTION

N°	Name	Society	Authorized copy

Authors: Corinne TERI
Christophe AILLAUD

UPDATES

Release	Date (dd/mm/yy)	Author	Modification
_01	04/04/01	C. Aillaud, C. Teri	Creation.
A00	15/02/02	C. Teri	Modifications according to the remarks included in TUV IT and BSI reports. Certificate ref: BSI-DSZ-CC-0187-2002.
A00P	19/02/02	C. Teri	Public Security Target.

TABLE OF CONTENTS

Authors: Corinne TERI
Christophe AILLAUD

LIST OF TABLES

LIST OF FIGURES

1. ST INTRODUCTION

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide document management and overview information such as labeling and descriptive information necessary to control and identify the ST and the TOE to which it refers, narrative form ST summary and state of any evaluatable claim of CC conformance for the TOE.

1.1 ST identification

Title:	ASE - Security Target
Reference:	**DPC102590**
Version:	**A00P**
Date of modification:	19/02/02
TOE:	**Java Card Platform Embedded Software**
TOE version:	**V3 (Core)**
Product:	GemXplore'Xpresso V3
IT Security scheme:	German scheme
Evaluation body:	TUV Informationstechnik GmbH evaluation body
Certification body:	Bundesamt für Sicherheit in der Informationstechnik (BSI)

This ST has been built with Common Criteria Version 2.1 (ISO 15408).

1.2 ST overview

The aim of this document is to describe the Security Target (ST) of the "**Java Card Platform Embedded Software**".

The product is GEMPLUS Java Card Platform Embedded Software (JCP ES) on a Smart Card Integrated Circuit (IC).

This product is based on the Smart Card IC to manage and execute Java Applications.

GemXplore'Xpresso, is a standard and interoperable solution for mobile services. GemXplore'Xpresso is the most comprehensive Java Card SIM solution available on the market today. It will drive the deployment of new mobile services through the highest level of interoperability with other Java Card 2.1 SIMs.

Mobile operators are increasingly turning to SIM-based services in order to differentiate themselves from the competition. Thanks to a unique set of features and an unbeatable efficiency for service delivery, the SIM Card has become the favorite platform for mobile services.

The main objectives of this ST are:

- To describe the Target-Of-Evaluation (TOE) as a card for a JCP ES.
- To define the limits of the TOE.
- To describe the security requirements for the TOE.

1.3 CC conformance claim

This ST is in accordance with the Common Criteria Version 2.1 (ISO 15408):

- Part 2 [**CCPART2**] extended.
- and Part 3 [**CCPART3**] conformant .

The minimum strength level for the TOE security functions is **SOF-high**.

The assurance level is **EAL4**.

2. TOE DESCRIPTION

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the TOE description as an assistance to the understanding of its security requirements, an addressing to the product or the system type and, a TOE's scope and boundaries general terms description.

2.1 TOE abstract

The Product under evaluation is the **GemXplore'Xpresso V3** card.

The TOE is the **Java Card Platform Embedded Software**.

The Java Card Platform Embedded Software (JCP ES) is a **Smart Card Embedded Software** that provides an **operating system** (OS) for GSM applications written in Java that can be hosted on a certified Smart Card Integrated Circuit (IC) with comparable level to the current TOE evaluation.

It is based on:

- The Java Card specification (see [**JCAPI, JCVM, JCRE**]);
- The Open Platform specification (see [**OP**]);
- The Visa Open Platform specification (see [**VOP**]) in compact configuration with PK (see [**OP2**]);

It uses:

- The certified chip's security requirements for the ES (see certification report **ITSEC E4 High** of Infineon **SLE66CX640P mask no-M1422a19** of chip for more details).

These de facto standards are aimed at defining a framework with which Applications can be developed, managed and used on a JCP ES.

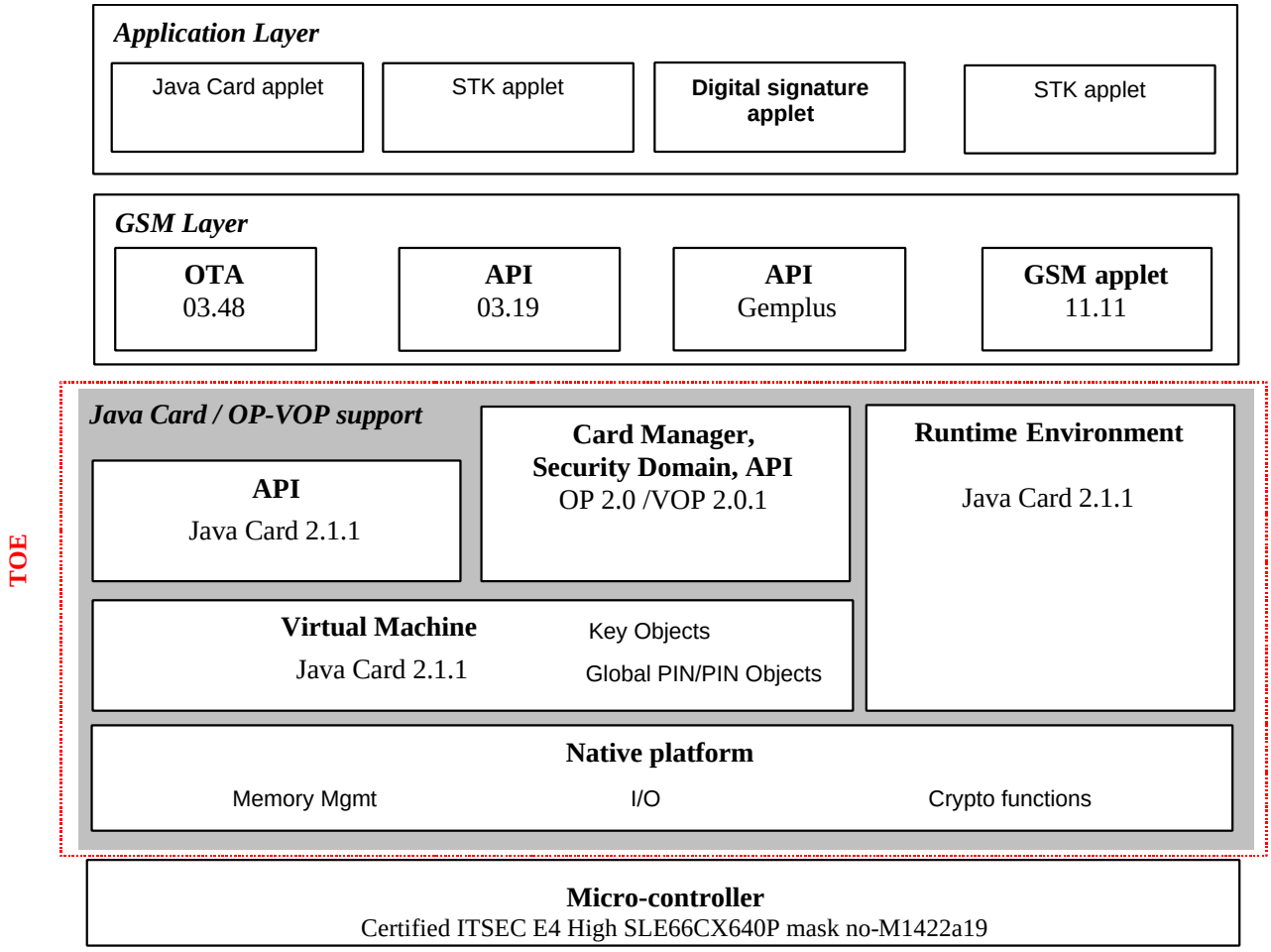


Figure 1 – Java Card Platform Embedded Software architecture

This figure shows the global architecture of the **Java Card Platform Embedded Software**.

The TOE includes all the Java Card / OP-VOP support modules and the native platform. **Each TOE module under evaluation (inside redline & on grey box in figure 1) is developed by GEMPLUS and based on the previous specified specifications.**

The TOE does not include the micro-controller (but used the certified chip's security requirements), the GSM layer and the Application.

Note: Due to the definition of the TOE, it is mandatory to define the physical environment – The micro-controller – on which the TOE is lying. The TOE uses information provided by the micro-controller to detect attacks.

2.2 TOE services

2.2.1 TOE ACTORS

2.2.1.1 Administrators

The description of the TOE administrators is given in the table below:

Administrator	Description
Product developer	The *Product developer* designs the chip ES. There the *Product developer* is GEMPLUS.
IC manufacturer	The *IC manufacturer* -or founder- designs, manufactures and loads the ES in the Smart Card IC. There the *IC manufacturer* is INFINEON.
Card manufacturer	The *Card manufacturer* is responsible for: • Manufacturing Smart Cards from the IC's provided by the *IC manufacturer*. • Loading and instantiating the JCP ES and Applications on the card. • Loading the JCP ES secrets, such as cryptographic keys and PIN. For this product, the *Card manufacturer* is GEMPLUS.
Personalizer	The *Personalizer* personalizes the card by loading the *Card issuer* and *End user* data as well as Application secrets such as cryptographic keys and PIN. For this product, the *Personalizer* is GEMPLUS.
Card issuer	The *Card issuer* -short named « issuer » issues cards to its customers that are the « *End users* ». The card belongs to the *Card issuer*. Therefore, the *Card issuer* is responsible for: • Selecting and managing the Applications. • Personalization the Applications. • Distribution the Applications. • Invalidation the Applications. For this product, the *Card issuer* is GSM operator.

Table 1 – TOE administrators

2.2.1.2 Users

The description of the TOE users is given in the table below:

User	Description
Application developer	The *Application developer* designs and implements the Applications that will be hosted on the Smart Card IC.

	For this product the *Application developer* is GEMPLUS.
End user	The *End user* (or cardholder) is a customer of the *Card issuer*. The card is personalized with the *End user* identification and secrets. He uses his personalized card with the his identification and secrets.
Terminals	*Terminal* equipment or card reader like Automatic Teller Machine (ATM), Point-Of-Sales terminal (POS), vending machines or Telephonic Mobile Equipment (ME). For this product the *Terminal* is the Telephonic Mobile Equipment.

Table 2 – TOE users

2.2.2 THE AIM OF THE TOE

The TOE is aimed to fight the following risks:

- Confidential data disclosure: Disclosure of confidential data in programmed microchip, i.e. Application code, keys, PIN.
- Identity usurpation: Management (i.e. load, personalization) of JCP ES and Application by unauthorized administrator, i.e. other than *Card manufacturer*, *Personalizer*, and *Card issuer*. Use of Application by unauthorized user, i.e. other than *End user*, and *Card issuer*.
- Data integrity loss: Use of a non-valid asset data.

2.2.3 CONTRIBUTION OF THE TOE IN THE APPLICATION

The TOE contributes to the Application by providing the following mechanisms:

- Logical separation or sharing of user data between Applications.
- Authentication of the TOE administrators.
- Confidentiality of the platform's cryptographic keys, PIN, ES.
- Integrity of the platform's cryptographic keys, PIN, ES.

It also contributes by providing basic mechanisms that are listed below. It is the responsibility of the *Application developer* to use these basic mechanisms properly in their Applications:

- Authentication of the *End user*.
- Confidentiality of the Application's cryptographic keys, PIN, and code.
- Integrity of the Application's cryptographic keys, PIN, and code.
- External bi-directional communication protection against disclosure and corruption (secure messaging).

In the applet developed by the *Application developer*, Global PIN and/or PIN could be used.

The *End user* has to know the Global PIN to use the TOE and after that there are one or more extra PINs to:

- Build an authentication for two or more *End users.*
- Make an extra (second) authentication for some high sensitive Applications.

The TOE can only have one Global PIN but many (one or more) PINs.

2.3 TOE life cycle

2.3.1 LIFE CYCLE

The Smart Card life cycle is composed of 7 phases.

However, due to the specificity of the JCP ES, we identify a new authority, the *Application developer*, that is in charge of designing and implementing the Application. The *Application developer* develops an applet which rely on the security mechanisms offered by the JCP ES as data's confidentiality and integrity (see the TOE services chapter).

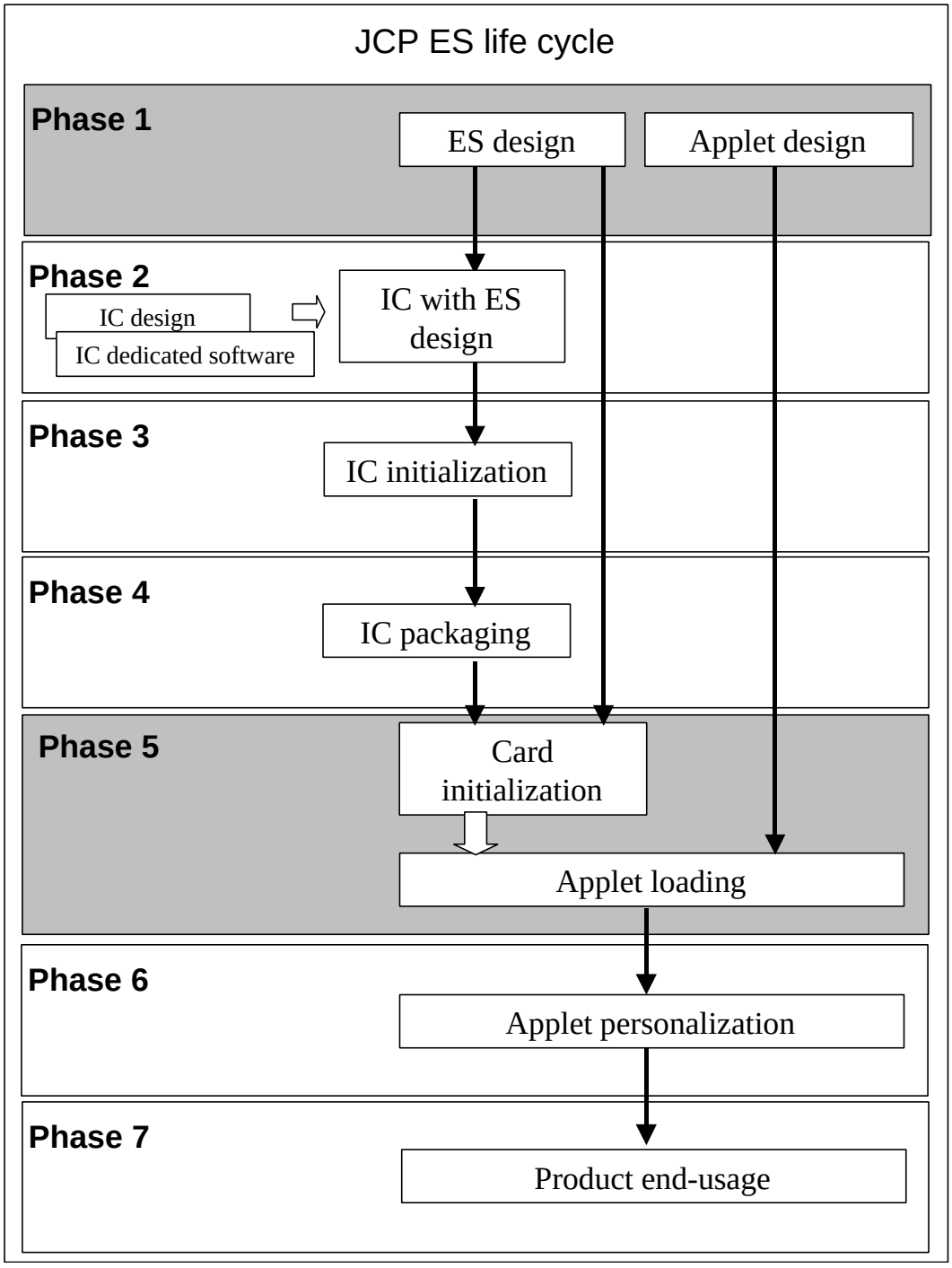


Figure 2 – JCP ES Life Cycle

According to the **Figure 2 – JCP ES Life Cycle**, the TOE environment is defined as follow:

- Development environment corresponding to phases **1** and 2, including the development environment of the *Application developer*, and IC Photomask Fabrication environment corresponding to phase 2;
- Production environment corresponding to phases 3, 4 and **5**, including the integration of the JCP ES into the IC, and the test operations, and loading and instantiating of the JCP ES and Application code;
- Personalization environment corresponding to phase 6, including personalization and testing of the Smart Card with the user data;
- User environment corresponding to phase 7, including usage of Application and related data.

2.3.2 DETAILS

Phase	Limit of the TOE	Industrial Phase	Industrial Deliverables	Logical Phase	TOE Administrators	TOE Users	Card State
1	Construction	Development	ES	ES Design	*Product developer*		None
			Application	Applet Design		*Application developer*	None
2	Construction	Development	Hard mask set	Chip Manufacturing	*IC manufacturer*		None
3	Construction	Production	Wafers with Chips	Chip Initialization	*IC manufacturer*		OS_NATIF
4	Construction	User – Production	Modules	Card Manufacturing	*Card manufacturer*		OS_NATIF
5	Construction	User – Production	Card with ES	Card Initialization (CM loading)	*Card manufacturer*		OP_READY INITIALIZED
			Card with application	Applet loading			SECURED
6	Usage	User – Personalization	Card personalized	Card Personalization	*Personalizer*		SECURED
7	Usage	User – Use		Card Distribution Card Termination	*Card issuer*	*End user* *Terminals*	SECURED

Table 3 – Smart Card phases

About phase 1:

The *Application developer* develops the applet to be loaded inside the card during the phase 5 and uses Java Compiler and Converter Virtual Machine in order to produce CAP and EXPORT files. Before loading these files inside the card, the *Card manufacturer* verifies them by using the SUN verifier off-card according to the "Java Card 2.1.2 off-card verifier" document [**JCVERIFIER**]. The role of this verifier is to check if CAP and EXPORT files are in conformance with the Java Card 2.1.1 specifications.

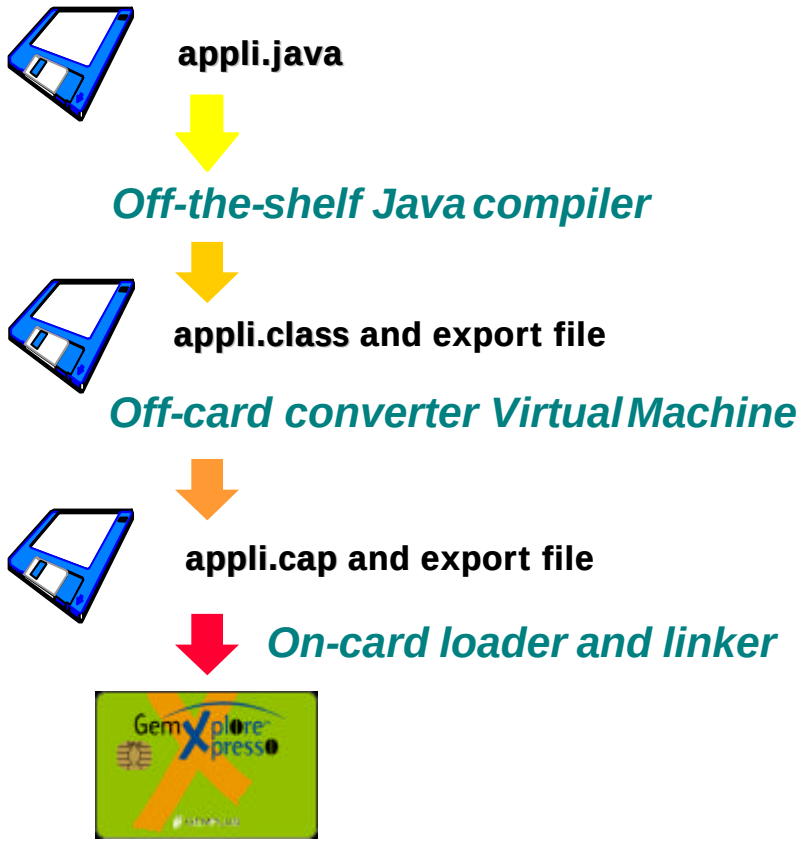


Figure 3 – Applet verification

About phase 5:

For the TOE "JCP ES", only one instance of the *OwnerPIN* class is created in order to be compliant with OP/VOP specifications. This particular instance is called the Global PIN.

Unfortunately the Product "GemXplore'Xpresso V3" doesn't use the TOE's *OwnerPIN* class instance. The GSM application uses his own PIN native implementation because more compact. This own PIN native implementation is outside the limit.

An other product (i.e. Banking product) uses the TOE's *OwnerPIN* class instance (i.e. Global PIN). If a loyalty application is inside the TOE with the banking application, then the loyalty application could use the same TOE's *OwnerPIN* class instance (i.e. Global PIN), or a new *OwnerPIN* class instance (i.e. PIN).

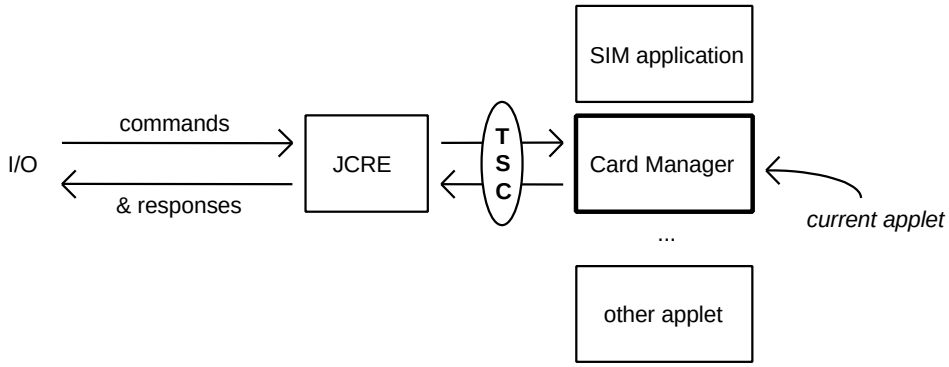
2.4 TOE intended usage

The TOE is an appropriate Embedded Software to implement the *Card issuer*'s policy in order to provide a JCP which can load, install, run and delete Java Card applications with different security levels.

The useful applications are Financial application (Credit/Debit, E-Purse, E-Commerce), Telephony application (SIM for mobile equipment), and E-signature application (Digital signature).

The *End user* uses the product by two modes: **connected** mode (inside the limit) and **OTA** mode (outside the limit).

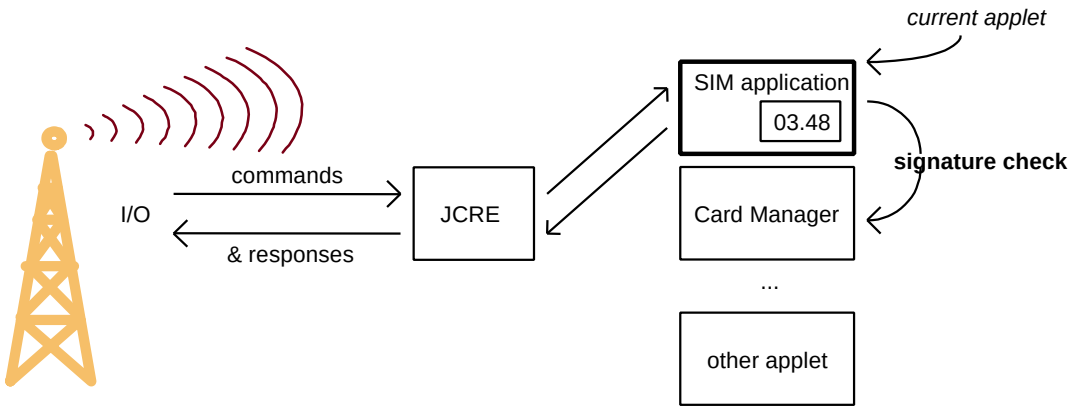
The connected mode is the following:



TSC = Trusted Secure Channel

The connected mode allows to use APDU commands (INSTALL, LOAD, DELETE, GET DATA, …) by I/O channel before personalization stage for administration usage, after post-issuance these commands shall only be used with OTA mode. Other APDU command (SELECT), API methods (OP, Java Card) and Application functions shall be used in connected mode.

The OTA mode is the following:



The OTA mode allows to download data or, GSM and OP APDU script (GSM Interpreter & Card Manager Interpreter) by physical layer (Envelope or Update Record SMS), transport layer (03.40), security layer (formatted 03.48 or unformatted messages), and application layer.

The GSM interpreter allows to execute APDU command via OTA by using the GSM specific TAR, APDU commands (CLA, INS, P1, P2, P3 + dataIn). Rights depends on access domain.

The OTA mode is outside the limit.

3. TOE SECURITY ENVIRONMENT

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the description of the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

The statement of the TOE security environment shall describe the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

This statement shall include the following:

- A description of assets
- A description of threats shall include all threats to the assets against which specific protection within the TOE or its environment is required. A threat shall be described in terms of identified threat agent, the attack and the asset that is the subject of the attack.
- A description of assumptions shall describe the security aspects of the environment in which the TOE will be used or is intended to be used.
- A description of organizational security policy shall identify, and if necessary explain, any organizational security policy statements or rules with which the TOE must comply.

3.1 Data objects (Assets)

3.1.1 PRIMARY ASSETS

D.APPLET	A piece of code executed by the TOE. This object has the following attribute: • The applet identifier (called AID).
D.GLOBAL_PIN	The **Card Manager** provide a mechanism for ***Card user* verification** that can be used by all applications on the card. The Open Platform provides for the implementation of a card Global PIN service in the Card Manager to support *Card user* verification requirements. The **D.GLOBAL_PIN** is an instance of the *OwnerPIN* class (defined in the Java Card specification) belonging to the TOE. The **D.GLOBAL_PIN** services allows to the ***Card user*** to : • **Update the D.GLOBAL_PIN**: sets a new value for the **D.GLOBAL_PIN** through an APDU command. The **D.GLOBAL_PIN** services allows to the **Applications** to: 1. **Check the D.GLOBAL_PIN**: compares the **D.GLOBAL_PIN** value against a presented value through a Java Card method. If the comparison is correct then the **D.GLOBAL_PIN** is validated else the **D.GLOBAL_PIN** is invalidated. 2. **Update the D.GLOBAL_PIN**: sets a new value for the **D.GLOBAL_PIN** through a Java Card method. This service is only available for privileged

	Applications.
D.PIN	The **TOE** provide a mechanism for *Card user* verification that can be used by all applications on the card. The **D.PIN** is an instance of the *OwnerPIN* class (defined in the Java Card specification) belonging to an Application. The **D.PIN** allows to its owner to : 1. **Check the D.PIN**: compares the **D.PIN** instance value against a presented value through a Java Card method. If the comparison is correct then the **D.PIN** is validated else the **D.PIN** is invalidated. 2. **Update the D.PIN**: sets a new value for the **D.PIN** and invalidates it through a Java Card method.
D.KEY	Set of Card Manager **(D.TSF_KEY)** and Application cryptographic **(D.USER_KEY)** keys used for Data Encryption Standard (DES) algorithm or Rivest, Shamir and Adleman Asymmetric ciphering algorithm (RSA).

3.1.2 SECONDARY ASSETS

D.BUFFERS	This entity is composed by two kinds of objets: **buffers in RAM** and **buffers in EEPROM**. • Buffers in RAM containing the data used for command processing and cryptographic computation. Command processing buffer (**D.APDU_BUFFER**) contains temporarily values of all the assets. Cryptographic computation buffer (**D.CRYPTO_BUFFER**) contains temporarily the value of the **D.KEY** assets. • Buffer in EEPROM containing the objects modified during the current transaction. This buffer, called transaction buffer (**D.TRANSACTION_BUFFER**), contains temporarily the value of the **D.GLOBAL_PIN** asset.
D.SECURE_CHANNEL	This entity corresponds to all the data transferred between TOE and the *Card user* in a secure way. This communication is achieved by a set of APDU commands.

3.2 Threats

3.2.1 THREAT AGENTS

S.OFFCARD	Attacker. A human or a process acting on his behalf being located outside the Smart Card IC. The main goal of the **S.OFFCARD** attacker is to access assets. Since the current evaluation is EAL4, the attacker has a high-level potential attack.

3.2.2 ATTACKS

T.CMD	The **S.OFFCARD** can use unauthorized instructions or commands or sequence of commands sent to the TOE in order to access the **D.APPLET,** the **D.GLOBAL_PIN** and the **D.KEY**.

T.IMPERSONATE	The **S.OFFCARD** can access the **D.APPLET**, the **D.GLOBAL_PIN**, the **D.PIN** and the **D.KEY** by an impersonalization mechanism.
T.LOAD_JCP	The **S.OFFCARD** can use unauthorized instructions or commands or sequence of commands sent to the TOE in order to modify the **D.APPLET**, the **D.GLOBAL_PIN**, the **D.KEY,** and the **D.BUFFERS**.
T.MOD_SHARE	The **S.OFFCARD** can modify the **D.APPLET** behavior by interacting on other **D.APPLET** in order to modify the **D.GLOBAL_PIN**, and the **D.KEY**.
T.LOAD_MAN	The **S.OFFCARD** can load a malicious Card Manager on the platform by using the card interface in order to access the **D.APPLET**, the **D.GLOBAL_PIN** and the **D.KEY**.
T.LOAD_APP	The **S.OFFCARD** can load **D.APPLET** on the platform by using the card interface in order to access and modify the **D.APPLET**, the **D.GLOBAL_PIN** and the **D.KEY**.
T.APP_DISC	The **S.OFFCARD** can intercept transmitted data in order to access and modify the **D.APPLET**, the **D.GLOBAL_PIN**, the **D.KEY** and the **D.SECURE_CHANNEL**.
T.APP_READ	The **S.OFFCARD** can use a malicious application by unauthorized mean in order to access and modify the **D.APPLET**, the **D.GLOBAL_PIN,** the **D.KEY** and the **D.PIN** belonging to another application.

3.3 Assumptions

A.CERTIFIED_CHIP	The chip shall be certified with comparable level to the current TOE evaluation. The chip to used by this JCP ES is the Infineon **SLE66CX640P mask no-M1422a19**. This chip is certified **ITSEC E4 High**. The main security features of the certified chip are the following: • Operating state checking. • Data encryption with on-chip key management and random number generation. • Phase management and test mode lock-out. • Protection against snooping.
A.CONVERTER	The converter shall generate verifiable Java Card bytecode, in a well-formed CAP file. The CAP file shall encapsulate the information contained in Java class files that comprise exactly one Java package. The package described in a CAP file shall define zero or more Java Card Applications (usually one). The converter shall check the limits imposed by the [**JC211**] specification on the number of classes, methods and fields. The converter shall only accept as input correct and consistent export files, and generate well-formed EXPORT files. The conversion process shall preserve the code semantics of the Application's Java code. At least access modifiers shall be correctly translated and the code correctly typed.
A.VERIFIER	The verifier shall verify individually each application before its loading on the card. The bytecode verifier shall assure that the bytecode instructions represent a legal set of Java instructions. Verification shall include testing that the bytecode is well-formed, overflow and underflow of stack frames, the correctness of parameters for all instructions, the correctness of all data conversions, the legality of accesses to private/public class members, and the validity of the register accesses and stores.

A.PINS_MGT	Only the *End user* shall know the **D.GLOBAL_PIN/D.PIN** code in a deciphered way. The **D.GLOBAL_PIN/D.PIN** code mailing shall be separate from the card mailing. A card shall never be close to any document giving **D.GLOBAL_PIN/D.PIN** contents. A third party like a GSM operator or an applet provider generates the **D.GLOBAL_PIN/D.PIN** code.
A.KEYS_MGT	The *Card issuer* and administrator servers shall keep all the JCP ES (**D.TSF_KEY**) and Application secret keys (**D.USER_KEY**) with a high level of confidentiality.
A.USE_SYS	It is assumed that the integrity and the confidentiality of assets stored/handled by the system (***Terminals***, communications...) shall be maintained.

3.4 Organizational security policies

As **there are no rules, procedures and practices imposed by organizations**, this chapter is not applicable to the TOE.

4. SECURITY OBJECTIVES

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the definition of the security objectives for the TOE and its environment. Security objectives address all the security environment aspects identified in the chapter above.

4.1 Security objectives for the TOE

OT.ID_AUT	The TOE shall ensure that **D.APPLET, D.GLOBAL_PIN, D.PIN, D.KEY,** and **D.BUFFERS** assets stored in memories are protected against any corruption or unauthorized disclosure and modification.
OT.ACCESS_CONTROL	The TOE shall ensure the separation between **D.APPLET** and data. The TOE shall ensure that a **D.APPLET** will not impersonate another **D.APPLET** to gain unauthorized accesses.
OT.ROLLBACK	The TOE shall ensure that in case of interruption of an operation through power failure or premature withdrawal of the card, it shall return all operational values to their status at the beginning of that operation.
OT.LOAD	The TOE shall ensure that the application can only be loaded and deleted via a **D.SECURE_CHANNEL**.
OT.DETECTIVE	The TOE shall ensure the detection of maximum number of failure attempts to open a secure channel or to get identified with the **D.GLOBAL_PIN/D.PIN**, is reached.
OT.INFO_PROTECTION	The TOE shall ensure that **D.BUFFERS** does not hold any usable information of the previous **D.APPLET** to the current **D.APPLET**.
OT.INTEGRITY_DETECTION	The TOE shall ensure the detection of an integrity error on the card life cycle state, **D.GLOBAL_PIN, D.PIN** and **D.KEY** assets.

4.2 Security objectives for the environment

OE.DEV_TOOLS	The environment shall ensure that the **D.APPLET** are verified, compiled, linked.
OE.USE_APPLICATION	The environment shall ensure that the **D.KEY** and the **D.GLOBAL_PIN/D.PIN** are kept secret even outside the TOE.
OE.USE_SYS	The environment shall ensure that the integrity and the confidentiality of **D.KEY** and **D.GLOBAL_PIN/D.PIN** assets handled by a ***Terminal*** are maintained.
OE.CERTIFIED_CHIP	The environment shall ensure that the TOE is implemented on a certified chip with comparable level to the current TOE evaluation. The chip used by this JCP ES is the Infineon **SLE66CX640P mask no-M1422a19**. This chip is certified **ITSEC E4 High**. The main security features of the certified chip are the following: • Operating state checking. • Data encryption with on-chip key management and random number generation.

	• Phase management and test mode lock-out. • Protection against snooping.
OE.CONFIDENTIALITY	The environment shall ensure that it is not possible to get the **D.KEY** and the **D.GLOBAL_PIN/D.PIN** from the *Card issuer*, the administrator and the *End user*.

5. IT SECURITY REQUIREMENTS

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the definition of the functional requirements for the TOE using only functional requirement components drawn from [**CCPART2**] and the definition of the assurance requirements for the TOE using only assurance components drawn from [**CCPART3**].

5.1 TOE security functional requirements

The TOE Security functional requirements define the functional requirements for the TOE using only functional requirement components drawn from [**CCPART2**].

The minimum strength level for the TOE security functions is **SOF-high**.

5.1.1 OBJECTS AND SUBJECTS

In this chapter, we will use the subjects and the objects defined in the following table.

S.CARD_MANAGER	The Card Manager is the subject that represents the *Card Issuer* in the card. It is a **D.APPLET** instance and also subject.
S.APPLET	All Java applets residing in the memories of the TOE. It is a **D.APPLET** instance and also a subject.
S.CIPHER	This subject is in charge of performing all cryptographic computations on the **D.KEY, D.PIN** and **D.GLOBAL_PIN** objects. Note: The D.PIN and D.GLOBAL_PIN are stocked by DES ciphering.
D.JAVA_OBJECT	Piece of data owned by an **S.APPLET** subject including specific data, initialization data, and personalization data.

We also need the definition of the some security attributes defined in the following table.

Object/Subject	Security attribute/Operation
D.APPLET	**Identifier**: This attribute corresponds to a universal identifier for the applet. **Load**: This operation corresponds to the loading of a new application on the TOE (LOAD APDU command). **Install**: This operation corresponds to the installation of an application on the TOE (INSTALL - for Install or Load - APDU command). **Delete**: This operation corresponds to the deletion of an application from the TOE (DELETE APDU command). **Select**: This operation corresponds to the selection of an application on the TOE (SELECT APDU command).
D.GLOBAL_PIN	**Ratification group**: This group is composed by the maximum presentation number and the retry counter.

	Security status: This attribute is a Boolean which indicates that the Global PIN has been correctly checked. **Update**: This operation corresponds to the update of the Global PIN by the administrator applet (PIN CHANGE UNBLOCK APDU command) or by a privileged applet (*OPSystem.setPin* API method). **Unblock**: This operation corresponds to the reset and unblock of the Global PIN by the administrator (PIN CHANGE UNBLOCK APDU command). **Check**: This operation corresponds to the check of the Global PIN by a privileged applet (*OPSystem.verifyPIN* API method).
D.PIN	**Ratification group**: This group is composed by the maximum presentation number and the retry counter. **Security status**: This attribute is a Boolean which indicates that the PIN has been correctly checked. **Update**: This operation corresponds to the update of the PIN by an applet (*OwnerPIN.update* API method). **Unblock**: This operation corresponds to the reset and unblock of the PIN by an applet (*OwnerPIN.resetAndUnblock* API method). **Check**: This operation corresponds to the check of the PIN by an applet (*OwnerPIN.check* API method). Note: D.PIN operations are submitted to firewall checks, which allow or deny an object access by an applet. See [JCRE] section 6 for more details.
D.KEY	**Type**: This attribute corresponds to the type of the cryptographic algorithm associated with the key. It defines also the key size. **Create**: This operation corresponds to the : - generation of the Key by an applet (*KeyBuilder.buildKey*, *KeyPair.genKeyPair* API methods). - loading of new Key on the TOE by the administrator (PUT KEY APDU command). **Delete**: This operation corresponds to the deletion of the Key (*Key.clearKey* API method). **Use**: This operation corresponds to: - decryption of the Key by an applet (*ProviderSecurityDomain.decryptVerifyKey* API method). - data ciphering or signing by an applet (*Cipher.update*, *Cipher.doFinal*, *Signature.sign*, *Signature.update*, *Signature.verify* API methods) **Update**: This operation corresponds to the update of the Key by an applet (*setKey*, *setModulus*, *setExponent*, *setP*, *setQ*, *setPQ*, *setDP1*, *setDQ1* API methods). **Read:** This operation corresponds to the Key value reading by an applet (*getKey*, *getModulus*, *getExponent*, *getP*, *getQ*, *getPQ*, *getDP1*, *getDQ1* API methods).
D.JAVA_OBJECT	**Owner**: This attribute defines the applet which owns the object.
D.SECURE_CHANNEL	**Ratification group**: This group is composed by the maximum presentation number

	and the retry counter. **Security status**: This attribute is a Boolean, which indicates that the secure channel has been correctly opened: i.e. the administrator has been authenticated.
S.CARD_MANAGER	**Identifier**: This attribute corresponds to a universal identifier for the Card Manager applet belonging to the TOE. **Life cycle state**: This attribute defines the state number of the card. According to this value, operations will available or not.
S.CIPHER	**Type**: This attribute corresponds to the type of the cryptographic key associated with the algorithm.

Table 4 – List of security attributes

TOE security functional requirements list

Component	Name
Security audit	
FAU_ARP.1	Security alarms
FAU_SAA.1	Potential violation analysis
Cryptographic support	
FCS_CKM.1	Cryptographic key generation
FCS_CKM.3	Cryptographic key access
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1	Cryptographic operations
User data protection	
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
FDP_DAU.1	Basic data authentication
FDP_ITC.1	Import of user data without security attributes
FDP_RIP.1	Subset residual information protection
FDP_ROL.1	Basic rollback
FDP_SDI.2	Stored data integrity monitoring and action
FDP_UCT.1	Basic data exchange confidentiality
Identification and authentication	
FIA_AFL.1	Basic authentication failure handling
FIA_ATD.1	User attribute definition
FIA_SOS.2	TSF generation of secrets
FIA_UAU.1	Timing of authentication
FIA_UAU.4	Single-use authentication mechanism

FIA_UID.1	Timing of identification
FIA_USB.1	User-subject binding
Security management	
FMT_MOF.1	Management of security functions behavior
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure security attributes
FMT_MSA.3	Static attribute initialization
FMT_MTD.1	Management of TSF data
FMT_MTD.2	Management of limits of TSF data
FMT_SMR.1	Security roles
Protection of the TSF	
FPT_FLS.1	Failure with preservation of secure state
FPT_PHP.3	Resistance to physical attack
FPT_RCV.4	Function recovery
FPT_RVM.1	Non-bypassing of the TSP
FPT_SEP.1	TSF domain separation
FPT_TDC.1	Inter-TSF data consistency
Trusted path/channels	
FTP_ITC.1	Trusted channel

Table 5 – List of TOE security functional requirements

User data list

Identification	Description
D.APPLET	see **chapter 3.1.**
D.USER_KEY	see **chapter 3.1.**
D.PIN	see **chapter 3.1.**

Table 6 – List of user data

TSF data list

Identification	Description
D.GLOBAL_PIN	see **chapter 3.1.**
D.TSF_KEY	see **chapter 3.1.**
D.BUFFERS	see **chapter 3.1.**
D.SECURE_CHANNEL	see **chapter 3.1.**

Table 7 – List of TSF data

5.1.2 SECURITY AUDIT (FAU)

5.1.2.1 FAU_ARP.1 Security alarms

FAU_ARP.1.1 The TSF shall take **one of the following disruptive actions** upon detection of a potential security violation.

List of disruptive actions:

1. **Reset the card and clear all volatile memory.**
2. **Block the action that produced the security violation and throw an exception.**
3. **Terminate the card (after this action, the card will stays mute forever).**
4. **Mute the card.**

Refinement:

The security alarms are generated by the TOE (see **FAU_SAA.1/SOFT**) and the IC (see **FAU_SAA.1/HARD** in the chapter **5.3 Security requirements for the IT environment**).

5.1.2.2 FAU_SAA.1 Potential violation analysis

FAU_SAA.1.1/ SOFT The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2/ SOFT The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of **the following auditable events** known to indicate a potential security violation:

List of auditable events:

1. **Card Manager life cycle state inconsistency.**
2. **Corruption of checksumed objects.**
3. **Illegal access to the previously defined D.JAVA_OBJECT objects.**
4. **Unavailability of resources audited through the object allocation mechanism.**
5. **Abort of a transaction that covers an object creation.**

b) Any other rules: **none.**

5.1.3 CRYPTOGRAPHIC SUPPORT (FCS)

5.1.3.1 FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1/ RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **(RSA) for the generation of public keys** and specified cryptographic key sizes **of single (512 bits) or double**

length (1024 bits) that meet the following standards:

1. **[VOP] sections 5, 6 and 7.**

FCS_CKM.1.1/ DES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **DES or 3-DES for the generation of session keys** and specified cryptographic key sizes **of single (64 bits) and double (128 bits) or triple length (192 bits)** that meet the following standards:

1. **[VOP] sections 5, 6 and 7.**

Refinement:

The RSA and DES cryptographic key generation use the IC security functional requirement (see **FCS_RND.1/HARD** in the chapter **5.3 Security requirements for the IT environment**).

5.1.3.2 FCS_CKM.3 Cryptographic key access

FCS_CKM.3.1 The TSF shall perform **the cryptographic keys decryption** in accordance with a specified cryptographic key access method **(OP/VOP command and OP/VOP Java API)** that meets the following standards:

1. **[OP] sections 8 and 9.9.**
2. **[VOP] section 9.3.**

Refinement:

The methods for cryptographic key decryption are PUT KEY APDU command and *OPSystem.decryptVerifyKey* API method.

5.1.3.3 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **(Java Card API)** that meets the following standards:

1. **[JCAPI] Interface Key.**

Refinement:

The method for cryptographic key destruction is *Key.clearKey* API method.

5.1.3.4 FCS_COP.1 Cryptographic operations

FCS_COP.1.1/ RSA The TSF shall perform **the encryption and decryption operations** in accordance with a specified cryptographic algorithm **RSA (RSA)** and cryptographic key sizes **of 512 bits, 768 bits and 1024 bits** that meet the following standards: **None.**

FCS_COP.1.1/ DES The TSF shall perform **encryption and decryption operations** in accordance with a specified cryptographic algorithm **Data Encryption Standards (DES)** and cryptographic key sizes **of 64 bits (DES) and 128 bits, 192 bits (Triple-DES)** that meet the following standards: **None.**

Refinement:

The RSA and DES encryption/decryption operations use the IC security functional requirements (see FCS_COP.1/HARD RSA and FCS_COP.1/HARD DES in the chapter **5.3 Security requirements for the IT environment**).

5.1.4 USER DATA PROTECTION (FDP)

5.1.4.1 FDP_ACC.2 Complete access control

FDP_ACC.2.1/ INIT	The TSF shall enforce the **Initialization access control SFP** on **the card life cycle management in phase 7,** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ INIT	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
FDP_ACC.2.1/ APPLET	The TSF shall enforce the **Applet access control SFP** on **the S. APPLET subjects** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ APPLET	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
FDP_ACC.2.1/ JAVA_OBJECT	The TSF shall enforce the **Java Object access control SFP** on **the subjects S.APPLET and the objects D.JAVA_OBJECT** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ JAVA_OBJECT	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
FDP_ACC.2.1/ KEY	The TSF shall enforce the **Key access control SFP** on **the subjects S.APPLET and S.CIPHER and the object D.KEY** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ KEY	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
FDP_ACC.2.1/ GLOBAL_PIN	The TSF shall enforce the **Global PIN access control SFP** on **the subjects S.APPLET and S.CIPHER and the object D.GLOBAL_PIN** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/ GLOBAL_PIN	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
FDP_ACC.2.1/ PIN	The TSF shall enforce the **PIN access control SFP** on **the subjects S.APPLET and S.CIPHER and the object D.PIN** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/ PIN	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.4.2 FDP_ACF.1 Security Attribute based access control

FDP_ACF.1.1/ INIT	The TSF shall enforce the **Initialization access control SFP** to objects based on **the card life cycle state.** **Initialization access control SFP:** **1. This SFP controls all the operations dedicated to the card life cycle state transition.** **2. Only the administrator and privileged S.APPLET can set the card life cycle state.** **3. Initial card life cycle state corresponds to the installation of the S.CARD_MANAGER at a specified AID.**
FDP_ACF.1.2/ INIT	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **1. The administrator and privileged S.APPLET can set the card life cycle state to new state according to the OP specification.**
FDP_ACF.1.3/ INIT	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**
FDP_ACF.1.4/ INIT	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none.**
FDP_ACF.1.1/ APPLET	The TSF shall enforce the **Applet access control SFP** to objects based on **the card life cycle state, D.SECURE_CHANNEL security status, the currently selected S.APPLET identifier, and the S.CARD_MANAGER identifier.** **Applet access control SFP:** **1. This SFP controls the following operations: load, install, and delete of an S.APPLET.** **2. Only the administrator can load, install and delete an S.APPLET upon receipt of an appropriate command message.** **3. The loading, installation, and deletion of an S.APPLET is possible during phase 5.** **4. The identifier of a S.APPLET is set to a given value at load.** **5. D. SECURE_CHANNEL security status is unset at card reset and initially.**
FDP_ACF.1.2/ APPLET	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **1. The loading, installation or deletion of an S.APPLET is allowed only if the TOE life cycle phase is phase 5.** **2. The loading, installation or deletion of an S.APPLET is allowed only if**

the currently selected S.APPLET identifier is equal to S.CARD_MANAGER identifier.

3. **The S.CARD_MANAGER–can load, install or delete an S.APPLET only if the D.SECURE_CHANNEL security status is equal to "true".**
4. **No restriction is made for the selection of an S.APPLET.**

FDP_ACF.1.3/ APPLET

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.4/ APPLET

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.1/ JAVA_OBJECT

The TSF shall enforce the **Java Object access control SFP** to objects based on **the currently selected S.APPLET identifier, and the D.JAVA_OBJECT owner.**

Java Object access control SFP:

1. **This SFP controls the following operations: access of a D.JAVA_OBJECT by an S.APPLET.**
2. **All conditions defined in the [JCRE] section 6 should be verified.**
3. **D.JAVA_OBJECT owner is applet that has created the D.JAVA_OBJECT.**

FDP_ACF.1.2/ JAVA_OBJECT

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

1. **The access of the D.JAVA_OBJECT by an S.APPLET shall be allowed only if the rules defined in the [JCRE] section 6 are all verified.**

FDP_ACF.1.3/ JAVA_OBJECT

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.4/ JAVA_OBJECT

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.1/ KEY

The TSF shall enforce the **Key access control SFP** to objects based on **the S.CIPHER (algorithm) type, and D.KEY type.**

Key access control SFP:

1. **This SFP controls the following operations: create, delete, use, update and read of a key value stored in a D.KEY.**
2. **Use of a key by an algorithm is allowed only if they have the same type.**
3. **Use of a key is allowed only if it is initialized.**

FDP_ACF.1.2/ KEY

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

1. **An S.CIPHER can use a D.KEY only if the D.KEY type matches the S.CIPHER (algorithm) type.**

FDP_ACF.1.3/ KEY	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**.
FDP_ACF.1.4/ KEY	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**.
FDP_ACF.1.1/ GLOBAL_PIN	The TSF shall enforce the **Global PIN access control SFP** to objects based on **the S.CIPHER (algorithm) type, and D.GLOBAL_PIN ratification group and security status.** **Global PIN access control SFP:** **1. This SFP controls the following operations: update, unblock and check of the Global PIN value stored in the D.GLOBAL_PIN.** **2. No user should read D.GLOBAL_PIN value.** **3. D.GLOBAL_PIN value update by a S.APPLET is allowed only if the S.APPLET has the associated privilege.** **4. The administrator can unblock and update the D.GLOBAL_PIN.** **5. Initial and maximum value of the D.GLOBAL_PIN ratification group is set at creation.** **6. D.GLOBAL_PIN security status is unset at card reset and initially.**
FDP_ACF.1.2/ GOBAL_PIN	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **1. An S.APPLET can check the D.GLOBAL_PIN only if the D.GLOBAL_PIN ratification group does not indicate that it is blocked.**
FDP_ACF.1.3/ GLOBAL_PIN	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**.
FDP_ACF.1.4/ GLOBAL_PIN	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **1. No S.APPLET shall have read access to the D.GLOBAL_PIN value.**
FDP_ACF.1.1/ PIN	The TSF shall enforce the **PIN access control SFP** to objects based on **the S.CIPHER (algorithm) type, and D.PIN ratification group and security status.** **PIN access control SFP:** **1. This SFP controls the following operations: update, unblock and check of the PIN value stored in the D.PIN.** **2. No user should read D.PIN value.** **3. An access (unblock, check, or update) to the D.PIN by an applet, is allowed if it fulfils the FDP_ACC/JAVA_OBJECT requirement.** **4. Initial and maximum value of the D.PIN ratification group is set at creation.**

5. **PIN security status is unset at card reset and initially.**

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

1. **An S.APPLET can check the D.PIN only if the D.PIN ratification group does not indicate that it is blocked.**

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

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

1. **No S.APPLET shall have read access to the D.PIN value.**

5.1.4.3 FDP_DAU.1 Basic Data Authentication

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **D.TSF_KEY, D.GLOBAL_PIN and D.PIN objects**.

FDP_DAU.1.2 The TSF shall provide **the S.CARD_MANAGER** with the ability to verify evidence of the validity of the indicated information.

5.1.4.4 FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the **Applet access control SFP and Java Object access control SFP** when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: **none.**

5.1.4.5 FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource from** the following objects:

- **D.JAVA_OBJECT.**

5.1.4.6 FDP_ROL.1 Basic rollback

FDP_ROL.1.1/ JAVA_OBJECT The TSF shall enforce **Java Object access control SFP** to permit the rollback of the **creation and the modification** on the **D.JAVA_OBJECT objects**.

FDP_ROL.1.2/ JAVA_OBJECT The TSF shall permit operations to be rolled back within the **boundary limit of the task being performed when operation is prematurely terminated**.

FDP_ROL.1.1/ KEY	The TSF shall enforce **Key access control SFP** to permit the rollback of the **loading** on the **D.KEY objects**.
FDP_ROL.1.2/ KEY	The TSF shall permit operations to be rolled back within the **boundary limit of the task being performed when operation is prematurely terminated**.

5.1.4.7 FDP_SDI.2 Stored data integrity monitoring and action

FDP_SDI.2.1/ KEY	The TSF shall monitor user data stored within the TSC for **integrity error** on all objects, based on the following attributes: **1. D.KEY value** **2. D.KEY object**
FDP_SDI.2.2/ KEY	Upon detection of a data integrity error, the TSF shall **deny the use of the corrupted D.KEY and:** **1. Mute the card if a D.KEY value integrity error is detected** **2. Thrown an exception if a D.KEY object integrity error is detected**
FDP_SDI.2.1/ GLOBAL_PIN	The TSF shall monitor user data stored within the TSC for **integrity error** on all objects, based on the following attributes: **1. D.GLOBAL_PIN value** **2. D.GLOBAL_PIN object**
FDP_SDI.2.2/ GLOBAL_PIN	Upon detection of a data integrity error, the TSF shall **deny the use of the corrupted D.GLOBAL_PIN and:** **1. Return false if a D.GLOBAL_PIN value integrity error is detected** **2. Thrown an exception if a D.GLOBAL_PIN object integrity error is detected**
FDP_SDI.2.1/ PIN	The TSF shall monitor user data stored within the TSC for **integrity error** on all objects, based on the following attributes: **1. D.PIN value** **2. D.PIN object**
FDP_SDI.2.2/ PIN	Upon detection of a data integrity error, the TSF shall **deny the use of the corrupted D.PIN and:** **1. Return false if a D.PIN value integrity error is detected** **2. Thrown an exception if a D.PIN object integrity error is detected**
FDP_SDI.2.1/ LOCK	The TSF shall monitor user data stored within the TSC for **integrity error** on all objects, based on the following attributes: **1. Card life cycle state value.**
FDP_SDI.2.2/	Upon detection of a data integrity error, the TSF shall **terminate the card.**

LOCK

5.1.4.8 FDP_UCT.1 Basic data exchange confidentiality

FDP_UCT.1.1	The TSF shall enforce the **Applet access control SFP, Key access control SFP and Global PIN access control SFP** to be able to **transmit and receive** objects in a manner protected from unauthorized disclosure.

5.1.5 IDENTIFICATION AND AUTHENTICATION (FIA)

5.1.5.1 FIA_AFL.1 Basic authentication failure handling

FIA_AFL.1.1/ APPLET	The TSF shall detect when **3** unsuccessful authentication attempts occur related to **any administrator authentication.**
FIA_AFL.1.2/ APPLET	When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **return an error.**

Refinement:

To authenticate the administrator, the cryptographic challenge/response protocol is used by INITIALIZE UPDATE and EXTERNAL AUTHENTICATE APDU commands. In FIA_AFL.1/APPLET, if the authentication fails then the card returns an error (i.e. it's impossible for administrator to get authenticated by the card).

FIA_AFL.1.1/ GLOBAL_PIN	The TSF shall detect when **a predefined number of** unsuccessful authentication attempts occur related to **any *End user* authentication.**
FIA_AFL.1.2/ GLOBAL_PIN	When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **block the D.GLOBAL_PIN.**

Refinement:

The predefined number of unsuccessful authentication is initially defined during the Card Manager initialization - between 3 to 15 (default: 10) - when the D.GLOBAL_PIN object is created.

To authenticate the *End user*, the Global PIN verification mechanism is used. In FIA_AFL.1/GLOBAL_PIN, if the authentication fails then the Global PIN is blocked (i.e. to unblock the Global PIN, only the administrator should use the PIN CHANGE UNBLOCK APDU command).

FIA_AFL.1.1/ PIN	The TSF shall detect when **a predefined number of** unsuccessful authentication attempts occur related to **any *End user* authentication.**
FIA_AFL.1.2/ PIN	When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **block the D.PIN.**

Refinement:

The predefined number of unsuccessful authentication is initially defined by the applet when the D.PIN object is created.

To authenticate the *End user*, the PIN verification mechanism is used. In FIA_AFL.1/PIN, if the authentication fails then the PIN is blocked (i.e. to unblock the PIN, the *OwnerPIN.resetAndUnblock* API method is used by the applet which has sufficient rights).

5.1.5.2 FIA_ATD.1 User attribute definition

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

1. **D.GLOBAL_PIN security status,**
2. **D.SECURE_CHANNEL security status.**

5.1.5.3 FIA_SOS.2 TSF generation of secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet **key length between 512 bits or 1024 bits for RSA keys and between 56 bits or 112 bits for DES keys.**

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for **the following TSF functions:**

1. **Cryptographic Key Management (SF_CRYPTO_KEY),**
2. **Secure Channel Management (SF_SECURE_MESSAGING).**

5.1.5.4 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow **the following TSF mediated actions** on behalf of the user to be performed before the user is authenticated.

TSF mediated actions list:

1. **Selection of an Application.**
2. **Recovery of S.CARD_MANAGER Data from the card.**
3. **Initiation of a D.SECURE_CHANNEL.**
4. **Execution of any command by the currently selected S.APPLET.**
5. **All actions which do not require user authentication.**

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.

5.1.5.5 FIA_UAU.4 Single-use authentication mechanisms

FIA_UAU.4.1/ APPLET The TSF shall prevent reuse of authentication data related to **the administrator authentication mechanism by using the one-time cryptographic challenge-response protocol.**

5.1.5.6 FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow the **execution of a S.APPLET** on behalf of the user **(*End user*)** to be performed before user **(*End user*)** is identified.

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

5.1.5.7 FIA_USB.1 User-subject binding

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

5.1.6 SECURITY MANAGEMENT (FMT)

5.1.6.1 Actions to be taken for management

Functions	**Actions**	**Applicable (A) / Not Applicable (NA)**
FAU_ARP.1	The management (addition, removal, or modification) of actions.	A
FAU_SAA.1	Maintenance of the rules by (adding, modifying, deletion) of rules from the set of rules.	NA
FCS_CKM.1 FCS_CKM.3 FCS_CKM.4	The management of changes to cryptographic key attributes (user, key_type, validity period, and use).	A A A
FCS_COP.1	No management.	-
FDP_ACC.2	No management.	-
FDP_ACF.1	Managing the attributes used to make explicit access or denial based decisions.	A
FDP_DAU.1	The assignment or modification of the objects for which data authentication may apply could be configurable in the system.	A
FDP_ITC.1	The modification of the additional control rules used for import.	A
FDP_RIP.1	The choice of when to perform residual information protection (i.e. upon allocation or de-allocation) could be made configurable within the TOE.	NA
FDP_ROL.1	Permission to perform a rollback operation could be restricted to a well-defined role.	A
FDP_SDI.2	The action to be taken upon the detection of an integrity error could be configurable.	NA
FDP_UCT.1	No management.	-
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts.	A
FIA_ATD.1	If so indicated in the assignment, the authorized administrator might be able to define additional security attributes for users.	A

FIA_SOS.2	The management of the metric used to generate the secrets.	A
FIA_UAU.1 FIA_UAU.4	Management of the authentication data by an administrator. No management.	A -
FIA_UID.1	The management of the users identities.	NA
FIA_USB.1	An authorized administrator can define default subject security attributes.	A
FMT_MOF.1	Managing the group of roles that can interact with the functions in the TSF.	A
FMT_MSA.1 FMT_MSA.2 FMT_MSA.3	Managing the group of roles that can interact with the security attributes. No management. Managing the group of roles that can specify initial values.	A - A
FMT_MTD.1	Managing the group of roles that can interact with the TSF data.	A
FMT_MTD.2	Managing the group of roles that can interact with the limits on the TSF data.	A
FMT_SMR.1	Managing the group of users that are part of a role.	NA
FPT_FLS.1	No management.	-
FPT_PHP.3	Management of the automatic responses to physical tampering.	NA
FPT_RCV.4	No management.	-
FPT_RVM.1	No management.	-
FPT_SEP.1	No management.	-
FPT_TDC.1	No management.	-
FTP_ITC.1	Configuring the actions that require trusted channel, if supported.	A

5.1.6.2 FMT_MOF.1 Management of security functions behavior

FMT_MOF.1.1 The TSF shall restrict the ability to **modify the behavior of** the functions **listed below** to **the Card issuer**.

1. **The management of the D. KEY.**
2. **The management of the D. GLOBAL_PIN.**
3. **The management of the D. PIN.**
4. **The management of the S.CARD_MANAGER life cycle.**
5. **The management of the loading, installation and deletion of an S.APPLET.**

5.1.6.3 FMT_MSA.1 Management of security attributes

FMT_MSA.1.1/OP The TSF shall enforce the **Applet access control, the Key access control, the Global PIN access control and the PIN access control SFPs** to restrict the ability to **perform the following operations on** the security attributes **defined below** to **the *Personalizer*, the *Card issuer* and the *End user* role.**

Object	Security attribute	Operation	SFP	Role

See Table 4	See Table 4	See Table 4	See FDP_ACC.2 And FDP_ACF.1	See FMT_SMR.1
D.APPLET	**Identifier**	**Load** **Install** **Delete**	**Applet access control**	***Personalizer*** **(phase 6)**
		Select	**Applet access control**	***End user*** **(phase 7)**
D.KEY	**Type**	**Create** **Delete** **Use** **Update** **Read**	**Key access control**	***Personalizer*** **(phase 6)**
				Card issuer **(phase 7)**
D.GLOBAL_PIN	**Ratification group**	**Update** **Unblock** **Check (*)**	**Global PIN access control**	***Personalizer*** **(phase 6)**
				Card issuer **(phase 7)**
D.PIN	**Ratification group**	**Update(*)** **Unblock(*)** **Check(*)**	**PIN access control**	***Personalizer*** **(phase 6)**
				Card issuer **(phase 7)**

Refinement:

A *user* is not able to operate directly on objects (**D.KEY**, **D.PIN**), but he should use an applet that performs it in order to operate on them.

(*) These operations can only be performed by an applet through API methods.

5.1.6.4 FMT_MSA.2 Secure security attributes

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

The secure value:

It is a value which security is assigned by all TSF requirements.

5.1.6.5 FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Initialization access control SFP, Applet access control SFP, Java Object access control SFP, Global PIN access control SFP and the PIN access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody** to specify alternative initial values to override the default values when an object or information is created.

Refinement:

For Initialization access control SFP, see FDP_ACF/INIT rule 3.

For Applet access control SFP, see FDP_ACF/APPLET rules 4 and 5.

For Java Object access control SFP, see FDP_ACF/JAVA_OBJECT rule 3.

For Global PIN access control SFP, see FDP_ACF/GLOBAL_PIN rules 5 and 6.

For PIN access control SFP, see FDP_ACF/PIN rules 5 and 6.

5.1.6.6 FMT_MTD.1 Management of TSF data

FMT_MTD.1.1/ KEY

The TSF shall restrict the ability to **access or modify** the **following TSF data** to **the *Card issuer* role (phase 7):**

1. **D.TSF_KEY.**

FMT_MTD.1.1/ GLOBAL_PIN

The TSF shall restrict the ability to **modify (in any way) by privileged applet** the **following TSF data** to **the *Card issuer* role (phase 7):**

1. **D.GLOBAL_PIN.**

5.1.6.7 FMT_MTD.2 Management of limits of TSF data

FMT_MTD.2.1

The TSF shall restrict the specification of the limits for **the following TSF data** to **the *Card manufacturer* (phase 5)**:

1. **D.GLOBAL_PIN retry counter.**
2. **D.SECURE_CHANNEL retry counter.**

FMT_MTD.2.2

The TSF shall take the following actions, if the TSF data are at, or exceed the indicated limits:

1. **For D.GLOBAL_PIN, return false.**
2. **For D.SECURE_CHANNEL, throw an error status word.**

5.1.6.8 FMT_SMR.1 Security roles

FMT_SMR.1.1

The TSF shall maintain the roles **defined in the following list.**

The roles list:

1. **The *Card manufacturer* role (phase 5).**

The *Card manufacturer* is in charge of initializing the secrets related to the JCP ES, and to set the Card Manager state to OP_READY, then INITIALIZED.

The *Card manufacturer* is in charge of setting the state to SECURED, once all Applications have been loaded and instantiated.

The *Card manufacturer* is in charge of loading the Application code load file into the Smart Card IC, and to set its state to LOADED.

The *Card manufacturer* is in charge of instantiating the Application code into an Application instance, and to set its state to INSTALLED, and then SELECTABLE.

The *Card manufacturer* is in charge of deleting:

- an Application if it doesn't shared any object

- or a Load file if it neither referenced by a file nor by an Application

2. The *Personalizer* role (phase 6).

The *Personalizer* is in charge of set the Applications' states to PERSONALIZED.

3. The *Card issuer* role (phase 7).

The *Card issuer* is in charge of managing the card life cycle.

4. The *End user* role (phase 7).

The *End user* is able to select an application.

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

5.1.7 PROTECTION OF THE TSF (FPT)

5.1.7.1 FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

1. **Card life cycle corruption.**
2. **Authentication data integrity failure.**
3. **Unexpected abortion of the execution of the TSF due to external events.**

5.1.7.2 FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1/ SOFT The TSF shall resist **the following physical tampering scenarios** to the **following TSF devices/elements** by responding automatically such that the TSP is not violated.

Devices/Elements	Physical tampering scenarios
Externally accessible interfaces	**Differential Power Analysis**

5.1.7.3 FPT_RCV.4 Function recovery

FPT_RCV.4.1 The TSF shall ensure that **all the SF's and failure scenarios (detailed in FPT_FLS.1)** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

5.1.7.4 FPT_RVM.1 Non-bypassing of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.7.5 FPT_SEP.1 TSF Domain separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2	The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.7.6 FPT_TDC.1 Inter-TSF data consistency

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **data types (defined in [VOP]) and S.APPLET code** when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use **the following interpretation rules** when interpreting the TSF data from another trusted IT product. **Interpretation rules list**: **1. The ISO 7816-6 rules [ISO7816].** **2. The [JCVM].**

5.1.8 TRUSTED PATH/CHANNELS (FTP)

5.1.8.1 FTP_ITC.1 Trusted channel

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and a remote 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	The TSF shall permit **remote users** to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for **D.APPLET loading, D.GLOBAL_PIN management, and D.TSF_KEY management.**

5.2 TOE security assurance requirements

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

The assurance level is **EAL4**.

TOE security assurance requirements list

Component	Name
Configuration management	
ACM_AUT.1	Partial CM automation
ACM_CAP.4	Generation support and acceptance procedures
ACM_SCP.2	Problem tracking CM coverage
Delivery and operation	
ADO_DEL.2	Detection of modification
ADO_IGS.1	Installation, generation, and start-up procedures

Development	
ADV_FSP.2	Fully defined external interfaces
ADV_HLD.2	Security enforcing high-level design
ADV_IMP.1	Subset of the implementation of the TSF
ADV_LLD.1	Descriptive low-level design
ADV_RCR.1	Informal correspondence demonstration
ADV_SPM.1	Informal TOE security policy model
Guidance document	
AGD_ADM.1	Administrator guidance
AGD_USR.1	User guidance
Life cycle	
ALC_DVS.1	Identification of security measures
ALC_LCD.1	Developer defined life-cycle model
ALC_TAT.1	Well-defined development tools
Tests	
ATE_COV.2	Analysis of coverage
ATE_DPT.1	Testing: high-level design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing – sample
Vulnerability assessment	
AVA_MSU.2	Validation of analysis
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.2	Independent vulnerability analysis

Table 8 – List of TOE security assurance requirements

5.3 Security requirements for the IT environment

This Chapter is closely linked to the micro-controller on which the TOE is lying and provides the Security requirements for the IT environment. Moreover, the TOE uses the certified chip's security requirements.

Security requirements for IT environment

Component	Name
Security audit	
FAU_SAA.1/HARD	Potential violation analysis
Cryptographic support	

FCS_COP.1/HARD	Cryptographic operation
FCS_RND.1/HARD	Quality metric for random numbers
Security management	
FMT_MSA.2/HARD	Secure security attributes
Protection of the TSF	
FPT_PHP.3/HARD	Resistance to physical attack

Table 9 – Security requirements for IT environment

Application note:

In this IT environment, the term **Smart Card IC** should replace the term **TSF**.

5.3.1 SECURITY AUDIT (FAU)

5.3.1.1 FAU_SAA.1 Potential violation analysis

FAU_SAA.1.1/ HARD The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2/ HARD The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of **the following auditable events** known to indicate a potential security violation:

List of auditable events:

1. **Frequencies out of range (low Frequency shall be greater than 800 kHz and high frequency shall be lower than 7.5 MHz).**
2. **Voltage out of range (low voltage shall be greater than 2.4 V and high voltage shall be lower than 6.2 V).**
3. **Temperature out of range (low temperature shall be greater than -25°C and high temperature shall be lower than 70°C).**

b) Any other rules: **none.**

5.3.2 CRYPTOGRAPHIC SUPORT (FCS)

5.3.2.1 FCS_COP.1 Cryptographic operation

FCS_COP.1.1/ HARD RSA The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **Data Encryption Algorithm (DEA) Rivest-Shamir-Adleman (RSA)** and cryptographic key sizes of **56 bit** that meet the following list of standards:

- **U.S. Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25, keying option 2**

- **ISO/IEC 9796-1, Annex A, sections A.4 and A.5, and Annex C**

FCS_COP.1.1/ HARD DES

The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **Data Encryption Algorithm (DEA) Data Encryption Standard (DES)** and cryptographic key sizes of **64 bit (56 for algorithm and 8 for parity)** that meet the following list of standards:

- **U.S. Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25, keying option 2**
- **ISO/IEC 9796-1, Annex A, sections A.4 and A.5, and Annex C**

5.3.2.2 FCS_RND.1 Quality metric for random numbers

FCS_RND.1/ HARD

The TSF shall provide a mechanism to generate random numbers that meet the following quality metric:

- **Generation in the RNGD (data) and RNGC (check) registers (8 bits)**
- **For RSA/DES keys generation**

5.3.3 SECURITY MANAGEMENT (FMT)

5.3.3.1 FMT_MSA.2 Secure security attributes

FMT_MSA.2.1/ HARD

The TSF shall ensure that only secure values are accepted for security attributes.

The secure value:

It is a value which security is assigned by all Smart Card IC requirements.

5.3.4 PROTECTION OF THE TSF (FPT)

5.3.4.1 FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1/ HARD

The TSF shall resist **the following physical tampering scenarios** to the **following TSF devices/elements** by responding automatically such that the TSP is not violated.

Devices/Elements	Physical tampering scenarios
Card life cycle state	**Erasure**
Clock	**Reduction of clock frequency to stop the TOE during a specific operation**
Clock	**Increase the clock frequency to corrupt TOE operation behavior**
Voltage supply	**Set voltage supply out of range**
Temperature	**Use the TOE in out of range temperature conditions to corrupt TOE operation behavior**

6. TOE SUMMARY SPECIFICATION

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the definition of the instantiation of the security requirements for the TOE and provide a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

6.1 TOE security functions

This chapter defines the list of the security functions for the TOE security functional requirements.

TOE security functions list

Function	Name
SF_ACCESS_CONTROL	TOE access control enforcement
SF_AUDIT	Security Audit
SF_CARD_TERMINATING	Card Life Cycle Management
SF_CRYPTO_KEY	Cryptographic Key Management
SF_CRYPTO_OPERATION	Cryptographic Computation
SF_IDENTIFICATION_AUTHENTICATION	End user Identification and Administrator Authentication
SF_INTEGRITY	Data Integrity
SF_PIN	PIN Management
SF_SECURE_MESSAGING	Secure channel Management
SF_TRANSACTION	Transaction Management

Table 10 – TOE security functions

6.1.1 SF_ACCESS_CONTROL

TOE access control enforcement

This security function is in charge of access control for the TOE. It is in charge of **Applet access control SFP (Applet loading, installation**, and **deletion)**, **Java Object access control SFP, Global PIN access control SFP, PIN access control SFP**, and **Initialization access control SFP (Card life cycle management).**

Concerning Applet access control (i.e. APDU commands privileges), the security function guarantees that:

- The only card user able to **load, install and delete** an **applet** is the administrator. This feature is only available during phase 5 of the TOE.
- The only card user able to **reset, unblock and change the Global PIN value** is the administrator.
- The only card user able to **set the card life cycle state** is the administrator.

- The only card user able to **load new key sets** is the administrator.

Concerning Java Object access control, the security function guarantees that:

- When a Java object access contravenes the access rules defined in the 6.2 section of the document [**JCRE**], this security function shall throw an exception.

Concerning Global PIN access control, the security function guarantees that:

- An Applet can not read the value of the Global PIN.
- An Applet can set a new value to the Global PIN only if it has the sufficient privileges.

Concerning PIN access control, the security function guarantees that:

- An Applet can not read the value of the PIN.
- PIN object access by an applet is submitted to the Java objects access control.

Concerning Initialization access control (i.e. Card life cycle management), the security function guarantees that:

- An Applet can lock the card only if it has the sufficient privileges.
- An Applet can terminate the card only if it has the sufficient privileges.

-

6.1.2 SF_AUDIT

Security Audit

This security function ensures the management of the following elements:

Element	Potential security violation	Automatic action
Hardware frequency	Frequencies out of range (low Frequency shall be greater than 800 kHz and high frequency shall be lower than 7.5 MHz)	Reset the card and clear all volatile memory.
Hardware voltage	Voltage out of range (low voltage shall be greater than 2.4 V and high voltage shall be lower than 6.2 V)	Reset the card and clear all volatile memory.
Hardware temperature	Temperature out of range (low temperature shall be greater than -25°C and high temperature shall be lower than 70°C)	Reset the card and clear all volatile memory.
Card Manager	Card Manager life cycle state inconsistency	Terminate the card (after this action, the card will stays mute forever).
Object	Abort of a transaction that covers Java object creation	Mute the card
Object	Corruption of checksumed objects	Block the action that produced the security violation and throw an exception.
D.JAVA_OBJECT	Illegal access to the previously defined D.JAVA_OBJECT objects	Block the action that produced the security violation and throw an exception.

Memory	Unavailability of resources audited through the object allocation mechanism.	Block the action that produced the security violation and throw an exception.

Table 11 – Security audit

6.1.3 SF_CARD_TERMINATING

Card Life Cycle Management

This security function ensures the management of the TOE life cycle:

- Only the administrator and privileged applets are able to change the card life cycle state.
- Only the administrator and privileged applets are able to obtain the card life cycle state.
- If the card life cycle state is corrupted, then the TOE is terminated.

6.1.4 SF_CRYPTO_KEY

Cryptographic Key Management

This security function controls all the operations relative to the cryptographic key management:

- Key generation:
  1. Automatic DES key generation manages 64, 128, 192 bits long keys.
  2. Automatic RSA key generation manages 512, 1024 bits long keys.
- Key decryption: the TOE provides Applications with a mean to decrypt keys which are imported using an APDU command. This service is provided by OP/VOP Java API.
- Key destruction: the TOE provides specified cryptographic key destruction methods that meet VOP standard.
- Key creation and update: the TOE provides specified key creation and modification methods.

6.1.5 SF_CRYPTO_OPERATION

Cryptographic Computation

This security function manages the cryptographic procedures provided by the TOE:

- A cryptographic algorithm must be initialized with a key that corresponds to its type and which length is correct before use.
- DES algorithm supports 64 bits, 128 bits 192 bits long keys.
- RSA algorithm supports 512 bits, 768 bits and 1024 bits long keys.
- The TOE provides a mean to generate a random number.
  1.
  2.
- A cryptographic algorithm cannot be processed if it has not been initialized.
- The TOE provides a mean to check the signature of data.

6.1.6 SF_IDENTIFICATION_AUTHENTICATION

End user Identification and Administrator Authentication

In this security function, we assume that the ***Terminal*** represents the administrator.

This security function ensures the management of the administrator authentication:

- The ***Terminal*** is authenticated through the administrator authentication mechanism, based on a one-time cryptographic challenge-response protocol.
- The administrator is the only card user able to open a secure channel.

This security function also manages the *End user* identification:

- The *End user* is identified through the Global PIN verification mechanism.
- Global PIN comparison with reference supplied by the *End user* for identification purpose. A retry counter associated to the Global PIN limits the number of attempts. The retry counter is decreased each time the identification fails. The Global PIN cannot be used for identification any longer if the retry counter reaches zero.

The strength of this function part is SOF-high.

6.1.7 SF_INTEGRITY

Data Integrity

This security function provides a mean to check the integrity of checksumed data stored in EEPROM: the Global PIN/PIN, the cryptographic keys, and the card life cycle state.

This security function initializes the checksum of an object at its creation.

6.1.8 SF_PIN

PIN Management

This security function controls the operations relative to a Global PIN/PIN management:

- Global PIN/PIN verification: a PIN can be accessed only if its format is correct.
- Global PIN/PIN modification: a PIN can be unblocked (reset the retry counter to the initial value) and changed (loading of a new value).
- Global PIN/PIN management: it is possible to manage (read, write) the validated flag, the retry counter of a PIN.

6.1.9 SF_SECURE_MESSAGING

Secure channel Management

This security function ensures the integrity and/or the confidentiality of command messages transmission in a secure channel. The integrity is achieved by adding a signature (Message Authentication Code: MAC) to the command message. The confidentiality is achieved by APDU message data field encryption. These features are used in accordance with the security mode applied to the secure channel.

Communication corruption: this security function guarantees the closing of the secure channel when it detects that the APDU are corrupted.

For this security function, the strength was not evaluated as it is a cryptographic algorithm suitable for encryption and decryption (See BSIG section 4, para. 3, clause 2).

6.1.10 SF_TRANSACTION

Transaction Management

This security function ensures the management of the transaction process. It provides assurance in the Java objects update in EEPROM:

- The content of the data that are modified within a transaction is copied in the transaction dedicated EEPROM area.
- Commit operation: closes the transaction, and clears the dedicated transaction area.
- Rollback operation: restores the original values of the objects (modified during the transaction) and clears the dedicated transaction area.
- The TOE manages an optimistic backup: the optimistic backup mechanism includes a backup of the previous data value at first data modification, and previous value restoring at abort.
- The security function ensures that the EEPROM containing sensitive data is in a coherent state whatever the time when EEPROM programming sequence stops, either during copying, invalidating, restoring data to or from the backup dedicated EEPROM area or updating sensitive data in EEPROM.

6.2 Assurance measures

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

Assurance measures list

Measure	Name
AM_ACM	Configuration management, reference ACM-DPC102594
AM_ADO	Delivery and Operation, reference ADO-DPC102595
AM_ADV	Development, reference ADV-DPC102598
AM_AGD	Guidance documents, reference AGD-DPC102591
AM_ALC	Life cycle, reference ALC-DPC102597
AM_ATE	Tests, reference ATE-DPC102603
AM_AVA	Vulnerability assessment, reference AVA-DPC102607

Table 12 – Assurance measures

6.2.1 AM_ACM: CONFIGURATION MANAGEMENT

This assurance measure ensures the configuration management. The CM responsible is in charge to write the CM plan, use the CM system and validate the CM system in order to confirm that ACM_XXX.Y components are completed.

6.2.2 AM_ADO: DELIVERY AND OPERATION

This assurance measure ensures the delivery and operation. The delivery responsible is in charge to write delivery documentation and validate it in order to confirm that the procedure is applied.

6.2.3 AM_ADV: DEVELOPMENT

This assurance measure ensures the development. The development responsible is in charge to design the TOE, write development documentation and validate it in order to confirm that the related security functional requirements are completed by security functions.

6.2.4 AM_AGD: GUIDANCE DOCUMENTS

This assurance measure ensures the guidance documents. The guidance responsible is in charge to write administrator and user guidance. The documentation provides the rules to use and administrate the TOE in a secured manner.

6.2.5 AM_ALC: LIFE CYCLE

This assurance measure ensures the life cycle. life cycle responsible is in charge to confirm that the life cycle process is applied.

6.2.6 AM_ATE: TESTS

This assurance measure ensures the tests. The test responsible is in charge to write tests and execute it in order to confirm that the security functions are tested.

6.2.7 AM_AVA: VULNERABILITY ASSESSMENT

This assurance measure ensures the vulnerability assessment. The security responsible is in charge to confirm that the security measures are suitable to meet the TOE security objectives conducing a vulnerability analysis.

7. PP CLAIMS

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide an optional claiming that the TOE conforms with the requirements of one, or more than one, PP.

This chapter is not applicable to this ST.

8. RATIONALE

OBJECTIVES OF THE CHAPTER

The objective of this chapter is to provide the evidence to be used for the ST evaluation and supporting the claims that the ST is a complete and cohesive set of requirements, that a conformant TOE would provide an effective set of IT security countermeasures within the security environment, that the TOE summary specification addresses the requirements and that any PP conformance claims are valid.

8.1 Security objectives rationale

The purpose of this chapter is to demonstrate the coverage of threats, assumptions and organizational security policies by the security objectives defined in the **chapter 3**.

This chapter is the GEMPLUS property.

8.2 IT security requirements rationale

The purpose of this chapter is to demonstrate the coverage of security objectives by the IT security requirements defined in the **chapter 5**.

This chapter is the GEMPLUS property.

8.3 TOE summary specification rationale

The purpose of this chapter is to demonstrate the coverage of security requirements by the security functions and assurance measures defined in the **chapter 6**.

This chapter is the GEMPLUS property.

8.4 PP claims rationale

This chapter is not applicable to this ST.

9. ABBREVIATIONS

See the **chapter Abbreviations** in the “References-Glossary-Abbreviations” document **[RGAR10030]**.

10. GLOSSARY

See the **chapter Glossary** in the “References-Glossary-Abbreviations” document **[RGAR10030]**.

11. REFERENCES

See the **chapter References** in the "References-Glossary-Abbreviations" document **[RGAR10030]**.

End of Document.