LEGIC advant® Series

Security Target Lite

LEGIC card-in-card AFS4096-JP12 V1.2

Common Critera ISO 15408 EAL4+

Classification: Public
Document No.: LA-23-615a-en / BSI-DSZ-CC-0812
Edition: 6.2012

www.legic.com

LEGIC Identsystems Ltd
Binzackerstrasse 41, CH-8620 Wetzikon, Switzerland
Phone +41 44 933 64 64, Fax +41 44 933 64 65
info@legic.com

Version Control

Version	Date	Changes to Previous Version	Remarks
1.2	2012-06-27	Initial public version	Revised by ZWA

Table of Contents

1. Introduction

1.1. ST Identification

Title:	Security Target LEGIC card-in-card AFS4096-JP12 V1.2
Version:	V1.0
Origin:	LEGIC Identsystems AG
Javacard OS platform:	NXP J3A081 (with JCOP 2.4.1 R3) [ZertJCOP081]
Security controller:	NXP P5CD081V1A (J3A081) [ZertIC081]
TOE identification:	LEGIC card-in-card AFS4096-JP12 V1.2
TOE documentation:	[Guidance1], [Guidance2], [Guidance3][1]

1.2. ST overview

The aim of this document is to describe the Security Target for a smart card (a security IC, usually integrated in a contactless smart card) with the LEGIC card-in-card applet. Thus, the TOE is a smart card comprising a hardware platform and a fixed software package. The software package is based on a Javacard operating system and the LEGIC card-in-card applet to manage the data stored in the non-volatile EEPROM memory. The specific function of this applet is to provide a secure memory area, which can be accessed only by legitimate readers. The specific TOE is based on a Javacard OS platforms (cf. section 1.1), which is certified according to CC EAL 5+ [ZertJCOP081]. The Javacard OS platform is based on a security controllers (cf. section 1.1) which are also certified according to CC EAL 5+ [ZertIC081].

The main objectives of this ST are:

- to introduce TOE and the according application,
- to define the scope of the TOE and its security features,
- to describe the security environment of the TOE, including the assets to be protected and the threats to be countered by the TOE and its environment during the product development, production and usage.
- to describe the security objectives of the TOE and its environment supporting in terms of integrity and confidentiality of application data and of protection of the TOE.
- to specify the security requirements which includes the TOE security functional requirements, the TOE assurance requirements and TOE security functions.

The assurance level for the TOE is CC EAL4, augmented with ALC_DVS.2 and AVA_VAN.5.

1.3. TOE overview

1.3.1. TOE definition

The Target of Evaluation (TOE) is a security IC (usually a contactless security token) providing a secured memory region usable by multiple applications.
The TOE consists of

[1] This document is only used by the LEGIC development department and is strictly confidential. Therefore it will not be part of the published ST Lite

- the circuitry of the smart card's chip (the integrated circuit, IC) with hardware for the contactless interface, e.g. antenna, capacitors (cf. section 1.1),
- the Javacard OS JCOP 2.4.1R3,
- the LEGIC card-in-card applet AFS4096-JP12 V1.2 as the only application on the platform described above,
- the associated guidance documentation: Preparational Guidance and Operational Guidance.

The TOE communicates through APDU with different applications (LEGIC reader). The APDU object is owned by the Java Card Runtime Environment (JCRE) and hence part of the certified Javacard platform. Communication between LEGIC reader and Applet running on the Security IC is done through APDU command/response pairs, while the reader always acts as master.

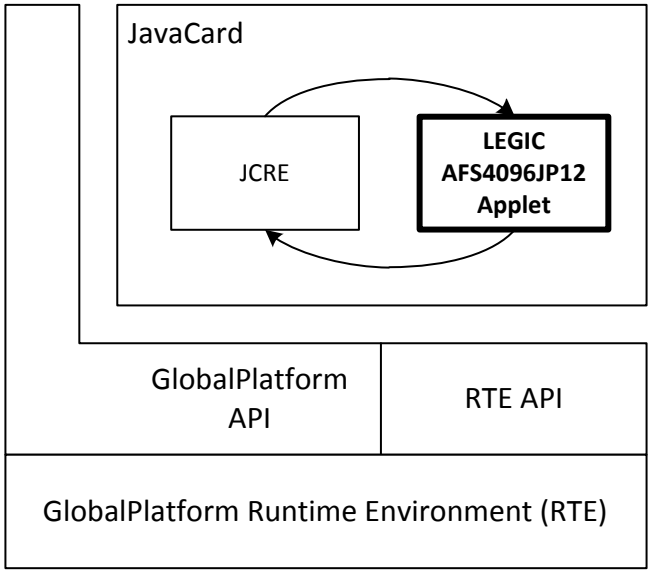


Figure 1 Basic software architecture of the TOE based on a Javacard operating system and the LEGIC card-in-card applet.

The entity to authenticate (the reader) has to prove the knowledge of the relevant secret keys in a Mutual Authenticate procedure. To protect message data against alteration and assure their authenticity, they are secured by a cryptographic checksum (MAC). The communication uses the Secure Messaging protocol defined in IEC/ISO7816-4.

1.3.2. TOE intended usage

Systems employing contactless smart cards as identification media intend to offer maximal user comfort in combination with high-level protection measures against a versatile range of attacks. Such systems use the smart card as a uniquely identifiable memory device capable to store any kind of data in a secure manner.
The content and the extent of the data filed on a smart card are dependent on the end-user application. The range of applications where smart cards, such as the TOE, are deployed as comfortable and well protected memory devices are manifold and range from physical and logical access control, time and attendance recording, transportation, cashless payment and access to leisure facilities.
In typical system configurations the smart card is assigned to a single individual user and hosts multiple applications, within dedicated memory areas of the TOE.
During the phase 6 of the outlined life cycle the TOE is used by the end-user. The method of use and the security requirements during this phase are dependent on the end-user

application. Independent on the end-user application the environment where the TOE is used during this phase isn't under the strict control of the System operator and must be assumed to be insecure. Therefore it must be anticipated that the TOE is exposed to various forms of attacks.
The TOE as an end-user device intends to support multiple applications, within the dedicated memory areas and to offer state-of-the-art counter measures against various attack scenarios. All life cycle phases of the TOE prior to the end-user phase are focused on the development and delivery of a user-friendly, high-end secured product enabling a broad field of applications while maintaining highest security measures.
Providers of smart card systems for dedicated application, such as access terminals, may use samples of the TOE during the development phase. These samples do not differ from the TOE and haven't got any additional functionality for testing purpose.
The TOE provides the following security functionalities:

- TSF_Access handles the enforcement of the access rights for transparent and record file access.
- TSF_Admin covers the process of initialization and personalization of the TOE.
- TSF_Secret ensures secure management of secrets such as cryptographic keys. It provides actions to read, update and use them.
- TSF_Crypto performs high level cryptographic operations. The implementation is based on the security functions provided by the Java Card platform.
- TSF_SecureMessaging provides the secure messaging: the MAC verification, the computation and the internal storage of the secure messaging session keys, and the optional encryption/decryption.
- TSF_Auth realizes the authentication mechanisms. It combines the possible authentication mechanisms and the storage of the authentication result.

In addition, TSF_Javacard represents the functionality of the underlying platform that is used by the applet. These functionalities are directly used during early phases like initialization and activation phase, but also for the realization of all other TOE security functionalities.

1.3.3. TOE life cycle

The life cycle of the TOE is described in terms of the following life cycle phases. Note that the life cycle definition focuses on the specific applet development and integration process. For the Javaoard Platform (NXP JCOP 2.4.1r3) as well as the underlying crypto library and security IC, different life cycle definitions have been used for their CC certification.

Nr.	Description
1	Software development This life cycle includes the development of the according applet, the testing as well as the guidance documentation for the TOE.
2	Software delivery After the software development, the applet software is delivered to the **Composite Product Manufacturer** being in charge for the composite product integration. The **Composite Product Manufacturer** may be LEGIC or an authorized and audited third party company. The security measures and the processes for the software delivery and storage at a **Composite Product Manufacturer** facility are defined and monitored through LEGIC.
3	Composite product integration and TOE delivery During this phase the **Composite Product Manufacturer** prepares the dedicated Javacards. This phase corresponds with phase 5 of [ZertJCOP]. This phase begins with the delivery of Javacards to the facility responsible for the

	composite product integration. The Composite Product Manufacturer then loads the encrypted applet to the card (where it is decrypted), finalizes the card and ships the card to the **Activation Agent**. Within this state (before the activation), the card isn't capable to be used as described in the intended use.

After the delivery of the TOE by the composite product integrator, the IC card typically runs through the following additional phases:

Nr.	Description
4	Activation During the activation the applet gets activated enabling the access to the applet through an authenticated terminal. The **Activation Agent** is authorized third party, which performs the activation according to the guidelines published by LEGIC. During the activation process no personal data are stored on the Smart Card. At the end of this phase the Smart Card is ready for the intended use and its internal operating state doesn't change until to the final disposal of the product. The card is then delivered to the Personalisation Agent.
5	Personalisation The **Personalisation Agent** is responsible for the applet personalisation of the smart card. From the TOE perspective this phase is identical to the intended operational use. The TOE is then delivered to the System Operator.
6	Operational Usage The **System Operator** is responsible for the smart card product delivery to the **end-user** and the appropriate management of the proceeding card life cycle until proper anonymisation, destruction and disposal of the smart card.

Note that these phases (4, 5 and 6) are not part of the TOE lifecycle according to this security target. Further the phases 4 and 5 may be done by the same agent.

Software development and software delivery

This life cycle includes the development of the according applet, the testing and the compilation of the guidance documentation for the TOE. The software development is done according to the guidance documentation of the certified javacard platform. The applet software is delivered to the composite product integration which usually takes place at a third party (the composite product manufacturer).
The security measures for the delivery to the composite product manufacturer and the storage of the software are defined and controlled by LEGIC.

Composite product integration and TOE delivery

The LEGIC card-in-card applet is installed on the Javacard platform IC. The installation is done through Global Platform commands and includes the following steps:

1. Mutual authentication between the init application and the card's issuer security domain using the Issuer security domain key
2. Uploading of the encrypted LEGIC applet
3. Deciphering of the LEGIC applet on the TOE.
4. Installation of the LEGIC applet and the transport key.
5. Finalisation of smart card according the operation guidance document of the underlying platform to ensure controlled smart card platform (disable installation of further software on the card; part of the initialisation of the card)
6. Delivery of the card with installed applet to an activation/personalisation line.

The composite product integration is depicted in the following scheme:

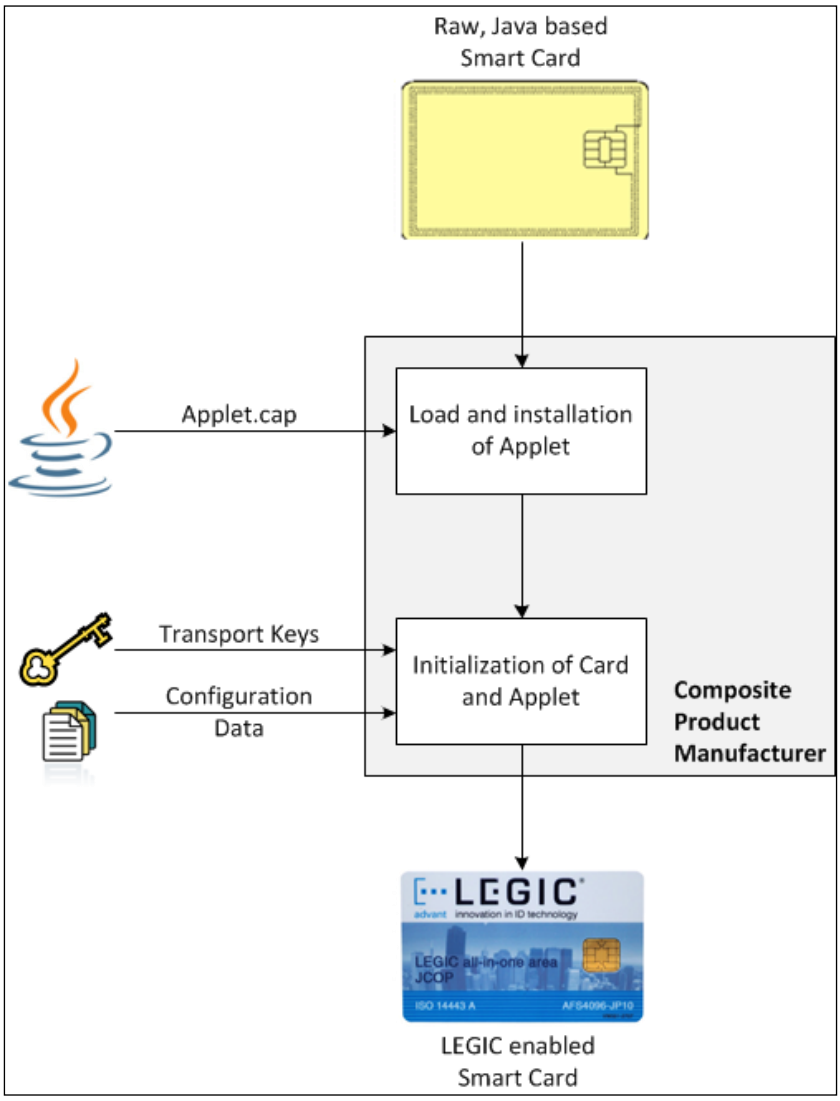


Figure 2: Scheme of the composite product integration. The composite product integration takes place at LEGIC or at a third party (the composite product manufacturer).

Installation and loading of the transport key must be executed in a secure environment. The security measures for the transport, storage and handling are defined, specified and controlled by LEGIC.

2. CC conformance

2.1. CC conformance claim

This security target claims conformance to:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, July 2009, version 3.1 revision 3, [CC_1],
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, July 2009, version 3.1 revision 3, [CC_2],
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, July 2009, version 3.1 revision 3, [CC_3],

as follows:

- Part 2 extended,
- Part 3 conformant,
- Package conformant to EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

The evaluation of the TOE uses the results of the CC evaluation of the chip platform; the chip platform claim conformance to the PP [PP0035]. The hardware part of the composite evaluation is covered by the certification report [ZertIC081]. In addition, the evaluation of the TOE uses the results of the CC evaluation of the JCOP Javacard OS claiming conformance to the PP [PP_Javacard]. The Javacard OS part of the composite evaluation is covered by the certification report [ZertJCOP081].

3. Security problem definition

This chapter is divided into the following sections: “Assets”, “Subjects”, “Threats”, “Organisational Security Policies” and “Assumptions”.

3.1. Assets

The central assets related to standard functionality of the TOE are:

- User data: the TOE provides a memory area which can be accessed by authenticated terminals. The data stored in this memory area by authenticated terminals are denoted “user data”.
- Cryptographic keys stored on the TOE [3]

While cryptographic keys are only written once during the activation process and cannot be read out, user data are read and written regularly by authenticated terminals during the intended use of the TOE.

To be able to protect these assets the TOE shall provide its security functionality. Therefore critical information about the TOE shall be protected.

3.2. Subjects

Within this security target, the following subjects are defined:

Subject name	Description
IC Deliverer	Usually the manufacturer of the security IC with integrated Javacard OS. Note that the IC deliverer is not part of the life cycles of the TOE as described within this security target.
Applet Manufacturer	The applet manufacturer is responsible for the management of the controlled product life cycle of the TOE [therefore covering the life cycle phases 1-3 of the defined life cycle of the TOE]. Therefore the he is responsible for the development, the maintenance and the distribution of the applet software.
Composite Product Manufacturer	The composite product manufacturer is responsible for the loading of the applet software to the Javacard security IC.
Terminal	Any system communicating with the TOE.

[3] Cryptographic keys are equal to TSF_DATA according to [CC_1, 4.1]

Subject name	Description
Authorized Terminal	A system communicating with the TOE which is authorized to read and write data to the TOE. Controlled by the corresponding authentication keys of the Terminal) an Authorized Terminal can have the security attribute “Activation Agent” or “LEGIC reader” with the corresponding access rights.
Attacker	A threat agent who tries to read or manipulate data (user data or cryptographic keys) stored within the TOE without authorization or forges a TOE. In general, every unauthorized subject trying to get access to these data is assumed to be an attacker with a potentially high experience level.
Activation Agent	The Activation Agent gets the TOE delivered from the composite product manufacturer. The Activation Agent is the organization, which replaces the transport key with the actual, card specific keys. Through this action the applet is activated and the access to the dedicated data objects through an authenticated terminal is possible. The activation process itself doesn’t contain any personalisation actions – therefore an activated applet holds no information required for end-user applications. Note that these actions are not part of the life cycle of the TOE as described within this security target.
Personalisation Agent	The Personalisation Agent gets the TOE delivered from the activation agent. [5] The Personalisation Agent is the organisation which deploys personalized data to a dedicated data object of the applet. This action is only possible on activated applets – therefore it’s not possible to store data on a not activated applet. Note that these actions are not part of the life cycle of the TOE as described within this security target.
System Operator	The System Operator is the authority using the smart card as measure for secure access control and identification within the environment under its control.
User	The user is the person, which uses the card as a data container for various applications. Note that user actions are not part of the life cycle of the TOE as described within this security target.

Table 1: List of subjects used in this security target.

3.3. Threats

The following threats are defined:

T.Data-Modification Unauthorised modification of data
User data stored by the TOE, written to the TOE or read from the TOE may be modified by unauthorised subjects.
Threat agent: Attacker

[5] The activation and the personalisation might be done by the same agent.

	Asset: User data Adverse action: Modification on the TOE without authorization, modification during read or write process.
T.Key-Eavesdropping	Eavesdropping during key transmission The transmission of keys written to the TOE by a legitimate reader may be eavesdropped by unauthorized subjects. Threat agent: Attacker Asset: Cryptographic keys Adverse action: Eavesdropping of the transmission.
T. Impersonate	Impersonating authorized users during authentication An unauthorized subject may try to impersonate an authorized subject during the authentication sequence, e.g. by a man-in-the middle or replay attack. Threat agent: Attacker Asset: User data (indirectly: access for modification after impersonation) Adverse action: Modification of the data transfer during communication with a authorized terminal, use of (e.g. replay) of parts of this communication with a non-authorized terminal.
T.Cloning	Cloning All or parts of the data stored on the TOE (including cryptographic keys) may be read out in order to create a duplicate of the TOE. Threat agent: Attacker Asset: Cryptographic keys (also user data, but they are not confidential) Adverse action: Reading of data by standard interface or by physical manipulation (probing).
T. Leakage	Information leakage An attacker may exploit information which is leaked from the TOE during usage of the Smart Card in order to disclose the confidential primary assets. This attack is non-invasive and requires no direct physical contact with the Smart Card Internals. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is the Differential Power Analysis (DPA). Another security concern is to take advantage of the susceptibility of the integrated circuit to put the TOE in an unsecured state. Threat agent: Attacker Asset: Cryptographic keys Adverse action: Side-channel attacks of various kinds (see above).
T.Fault	Fault attacks An attacker may cause a malfunction of TSF or of the Smart Card embedded software by applying environmental stress in order to (1) deactivate or modify security features or functions of the TOE or (2) deactivate or modify security functions of the Smart Card embedded

software. This may be achieved by operating the Smart Card outside the normal operating conditions.
Threat agent: Attacker
Asset: Cryptographic keys, user data (modification)
Adverse action: Causing malfunctions by radiation pulses, LFI or other means.

3.4. Organisational Security Policies

The following additional policies are defined in this Security Target:

P.Auth	Authentication The TOE shall enforce the authentication of the reader prior to any operations (e.g. read or write) on user or key data.
P.MAC	Integrity during communication The TOE shall enforce the use of integrity protected communication for all operations (e.g. read or write) on user data.
P.Encryption	Confidentiality during communication The TOE shall enforce the use of encrypted communication for all operations (write) on key data.

3.5. Assumptions

The following assumptions for the operational environment are made in this Security Target:

A.Secure_Keys	Usage of secure keys Only confidential and secure keys shall be used to set up the authentication and access rights. With the exception of session keys, key values are generated outside the TOE and downloaded to the TOE.[6]
A.Terminal_Support _Int	Terminal support to ensure integrity The terminal verifies information sent by the TOE in order to ensure the integrity of the communication.
A.Terminal_Support _Conf	Terminal support to ensure confidentiality For those transmission processes where confidentiality of transmitted data is an issue, the terminal encrypts information sent to the TOE in order to ensure the confidentiality of the communication.

[6] After the activation there are no key management, key update or key download functionalities in place.

4. Security objectives (ASE_OBJ)

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

4.1. Security Objectives for the TOE

This section describes the security objectives for the TOE addressing the aspects of identified threats to be countered by the TOE and organizational security policies to be met by the TOE. The main security objectives for the TOE are:

OT.Access-Control	Access Control The TOE must provide an access control mechanism for data stored by it. The access control mechanism shall apply to all operations for data elements. The cryptographic keys used for authentication shall never be output.
OT.Authentication	Authentication The TOE must provide an authentication mechanism in order to be able to authenticate authorised users. The authentication mechanism shall be resistant against replay and man-in-the-middle attacks.
OT.MAC	Integrity-protected Communication The TOE must be able to protect the communication by adding a MAC.
OT.Encryption	Encrypted Communication The TOE must be able to encrypt the communication.

In addition, the composite TOE must be resistant against various attacks and malfunctions (the objectives are mainly reached by security measures of the underlying Javacard platform (cf. the security target [ASE_JCOP081]):

OT.Protect_Data	Protection of stored data The TOE shall ensure that sensitive information stored in memories is protected against unauthorized disclosure and any corruption or unauthorized modification. Moreover, the TOE shall ensure that sensitive information stored in memories is protected against unauthorized access. The TOE has to provide appropriate security mechanisms to avoid fraudulent access to any sensitive data, such as cryptographic keys, authentication data or any other access controlled information.
OT.Side_Channel	The TOE must provide protection against disclosure of primary assets including confidential data (User Data or TSF data) stored and/or processed in the Smart Card IC: • by measurement and analysis of the shape and amplitude of the power consumption or electromagnetic emanation,

- by measurement and analysis of the time between events found by measuring signals (for example on the power, clock, or I/O lines).

Especially, the smart card embedded software must be designed to avoid interpretations of signals extracted, intentionally or not, from the hardware part of the TOE (for instance, Power Supply, Electro Magnetic emissions).

OT.Fault_Protect	The TOE must ensure its correct operation even outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include voltage, clock frequency, temperature, or external energy fields that can be applied on all interfaces of the TOE (physical or electrical).
OT.Physical	The TOE hardware provides the following protection against physical manipulation of the IC, and prevent • Reverse-engineering (understanding the design and its properties and functions), • Physical access to the IC active surface (probing) allowing unauthorized memory content disclosure, • Manipulation of the hardware security parts (e.g. sensors, cryptographic engine or RNG), • Manipulation of the IC, including the smart card embedded software and its application data (e.g. authentication flags, etc.).

4.2. Security Objectives for the Operational Environment

The following security objectives for the environment are defined in this Security Target:

OE.Secure_Keys	Generation of secure keys The environment shall generate confidential and secure keys for authentication and encryption purposes. With the exception of session keys, key values are generated outside the TOE and they are downloaded to the TOE during system integration and personalization.
OE.Terminal_Support_Int	Terminal support to ensure integrity The terminal shall verify information sent by the TOE in order to ensure the integrity of the communication. This involves checking of MAC values, verification of information according to the cryptographic protocol and secure closing of the communication session.
OE.Terminal_Support _Conf	Terminal support to ensure confidentiality For those transmission processes where confidentiality of transmitted data is an issue, the terminal shall encrypt information sent to the TOE in order to ensure the confidentiality of the communication.

5. Extended Components Definition

5.1. Definition of the Family FCS_RNG

To define the IT security functional requirements of the TOE a sensitive family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.
The family “Generation of random numbers (FCS_RNG)” is specified as follows.

FCS_RNG Generation of random numbers

Family 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 Generation of random numbers	1

FCS_RNG.1 Generation of random numbers requires that 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 **Quality metric for random numbers**

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

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

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

5.2. Definition of the Family FPT_EMSEC

The family FPT_EMSEC (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against the TOE and other secret data where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks are evaluation of TOE’s electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc. This family describes the functional requirements for the limitation of intelligible emanations which are not directly addressed by any other component of CC part 2 [CC_2].

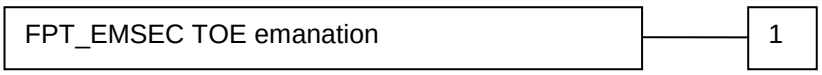
The family “TOE Emanation (FPT_EMSEC)” is specified as follows.

FPT_EMSEC TOE Emanation

Family behavior

This family defines requirements to mitigate intelligible emanations.

Component leveling:



FPT_EMSEC.1 TOE emanation has two constituents:

FPT_EMSEC.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.

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

Management: FPT_EMSEC.1
There are no management activities foreseen.

Audit: FPT_EMSEC.1
There are no actions defined to be auditable.

FPT_EMSEC.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

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

6. Security Requirements

6.1. Security Functional Requirements for the TOE

This section on security functional requirements for the TOE is divided into sub-sections following the main security functionality.
The text of the security functional requirements uses the following typographic notation:

- **Bold and underlined**: assignment.
- ***Bold and slanted***: selection.

6.1.1. Class Cryptographic Support (FCS)

FCS_COP.1/TDES	**Cryptographic operation – Symmetric Encryption / Decryption with TDES**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/TDES	The TSF shall perform **secure messaging – encryption and decryption** in accordance with a specified cryptographic algorithm: **Triple-DES in CBC mode** and cryptographic key size **112 bit** that meets the following: [**FIPS46-3**].

FCS_COP.1/MAC	**Cryptographic operation – 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] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/MAC	The TSF shall perform **secure messaging – message authentication code** in accordance with a specified cryptographic algorithm: **Retail-MAC** and cryptographic key size **112 bit** that meets the following: **[ISO9797-1]**.

FCS_CKM.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **(physically overwriting the keys with zeros)** that meets the following: [**none**].

Application note: This requirement is related to the platform requirement FCS_CKM.4.1 of [ASE_JCOP081].

FCS_RNG.1	**Quality metric for random numbers**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a **hybrid** random number generator that implements **a deterministic RNG according to ANSI X9.31 [ANSIX9.31].**
FCS_RNG.1.2	The TSF shall provide a mechanism to generate random numbers that meet **the AIS20 Class K3 quality metric [AIS20].**

Application note: This requirement is related to the platform requirement FCS_RNG.1 of [ASE_JCOP081].

6.1.2. Class Identification and Authentication (FIA)

FIA_UAU.1	**Timing of authentication**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.1.1	The TSF shall allow **reading of TOE identification and starting of the authentication protocol** 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.

FIA_UAU.3	**Unforgeable authentication**
	Hierarchical to: No other components. Dependencies: No dependencies.
FIA_UAU.3.1	The TSF shall **prevent** use of authentication data that has been forged by any user of the TSF.
FIA_UAU.3.2	The TSF shall **prevent** use of authentication data that has been copied from any other user of the TSF.

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

FIA_UID.1.1	The TSF shall allow **1. to read the unique TOE identification information** 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.

6.1.3. Class User Data Protection (FDP)

FDP_ITC.1/GP **Import of User Data without security attributes**

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation
FDP_ITC.1.1/GP	The TSF shall enforce the **access control SFP: Global platform mutual authentication** when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/GP	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/GP	The TSF shall enforce the following rules when importing user data c ontrolled under the SFP from outside the TOE: **(1) Setting the file system authentication keys and the transponder authentication key to the transport key after applet loading and initialisation is only possible after successful Global platform authentication and the setup of an encrypted and integrity-protected secure messaging channel.**

FDP_ITC.1/Act **Import of User Data without security attributes**

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation
FDP_ITC.1.1/Act	The TSF shall enforce the **access control SFP: LEGIC-specific mutual authentication** when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/Act	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/Act	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **(1) After delivery of the TOE, writing the file system authentication keys and the transponder authentication key is only possible after successful authentication with**

the transponder authentication key and the setup of an encrypted and integrity-protected secure messaging channel.[7]

FDP_UIT.1	**Data exchange integrity**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]
FDP_UIT.1.1	The TSF shall enforce the **Access Control SFP: LEGIC-specific mutual authentication with secure messaging to be able** to ***transmit and receive*** user data in a manner protected from ***modification, deletion, insertion and replay*** errors.
FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether ***modification, deletion, insertion and replay*** has occurred.
FDP_ACC.1	**Subset access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1	The TSF shall enforce the **Access Control SFP: LEGIC-specific mutual authentication** on **terminals gaining write, read and modification access to user data.**
FDP_ACF.1	**Security attribute based access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1	The TSF shall enforce the **Access Control SFP: LEGIC-specific mutual authentication** to objects based on the following: **1. Subjects:** **a. Terminal**[8]**,** **2. Objects:** **a. user data** **b. cryptographic keys** **3. Security attributes:** **a. authentication status the of terminal.**

[7] Note that in this phase the transponder authentication key is set to the transport key.

[8] The Terminal may be a general Terminal (any system communicating with the TOE) or an Authorized Terminal according to section 3.2.

FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **1. the successfully authenticated Terminal is allowed to read and write user data.**
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the rule: **1. A not successfully authenticated terminal is not allowed to read data or write user data,** **2. A not successfully authenticated terminal is not allowed to read data or write cryptographic keys.**

6.1.4. Class Protection of the Security Functions (FPT)

FPT_EMSEC.1	**TOE Emanation**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_EMSEC.1.1	The TOE shall not emit **variations in power consumption, electromagnetic emanation or timing during command execution** in excess of **non-useful information** enabling access **to cryptographic keys** and **user data**.
FPT_EMSEC.1.2	The TSF shall ensure **that unauthorized users** are unable to use the following interface: **contactless interface or direct physical access to the smart card IC** to gain access to **cryptographic keys** and **user data**.
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: • **Exposure to out-of-range operating conditions where therefore a malfunction could occur,** • **failure detected by TSF during run time.**
FPT_PHP.3	**Resistance to Physical Attack**[9]

[9] FPT_PHP.3 and the according refinement have been taken from the security target of the Java Card platforms [ASE_JCOP080], [ASE_JCOP081].

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

6.1.5. Class Security Management (FMT)

FMT_SMF.1	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No Dependencies
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: **1. Initialisation ,** **2. Activation** **3. Deactivation after fault detection.**

FMT_SMR.1	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles **1. Composite Product Manufacturer,** **2. Activation Agent,** **3. Authorized Terminal.**
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

6.2. Security Assurance Requirements for the TOE

The for the evaluation of the TOE and its development and operating environment are those taken from the

Evaluation Assurance Level 4 (EAL4)

and augmented by taking the following components:

ALC_DVS.2 and AVA_VAN.5.

7. TOE Summary Specification

7.1. Security Functionality

7.1.1. TSF_Access: Access rights

This security functionality handles the enforcement of the access rights for transparent and record file access. TSF_Access is employed whenever file structures (user data, cryptographic keys) are accessed. Its functionality is not concentrated but spread over several functions.
It combines the knowledge about the requested access mode, the file that is to be accessed and the current state of authentication in order to evaluate whether or not the access is granted.

7.1.2. TSF_Admin: Administration

This security functionality covers the process of initialization, activation and personalization of the TOE. Most of this functionality is directly provided by the underlying TSF_JavaCard (upload and installation of Applet, setting of the transport key). This security function includes also the storage of unique TOE identification information data.

7.1.3. TSF_Secret: Secret key management

This security functionality ensures secure management of secrets such as cryptographic keys. It provides actions to read, update and use them. Cryptographic keys are used for authentication, data encryption and MAC calculation/verification.

7.1.4. TSF_Crypto: Cryptographic operations

This security functionality performs high level cryptographic operations. The implementation is based on the security functions provided by TSF_Javacard. It supports secure messaging and is providing the service for de- and encryption of data or MAC calculations. For all cryptographic mechanism Triple-DES encryption/decryption or Triple-DES retail MAC is used (16-byte symmetric key with 112 independent key bits).

7.1.5. TSF_SecureMessaging: Secure Messaging

This security functionality provides the secure messaging, the MAC verification, the computation and the internal storage of the secure messaging session keys. It provides two kind of secure messaging: one only protecting the APDU integrity by a MAC, and one that also communicates with encrypted payload data.

7.1.6. TSF_Auth: Authentication protocols

This security functionality realizes the authentication mechanisms. TSF_Auth combines the possible authentication mechanisms and the storage of the authentication result. It also contains a module, where these results are stored for subsequent evaluation, whether or not one or the other authentication has been done successfully.
In order to process authentication protocols, this TSF may use the stored keys in the TOE. It provides access to two mutual authentication protocols: the proprietary LEGIC protocol and the GlobalPlatform algorithm provided by TSF_JavaCard.

7.1.7. TSF_Javacard: Javacard OS security functions

TSF_JavaCard represents the functionality of the underlying platform that is used by the applet. These functionalities are directly used during early phases like initialization and activation phase.
The Javacard operation system in general (part of the TOE) features the following TOE Security Functions. The exact description can be found in the Javacard OS security target ([ZertJCOP081]):

- Enforcement of access control (SF.AccessControl), used e.g. for TSF_Admin and TSF_Access.
- Audit functionality (SF.Audit)
- Cryptographic key management (SF.CryptoKey), used e.g. by TSF_Secret.
- Cryptographic operations (SF.CryptoOperation), used e.g. by TSF_Crypto.
- Identification and authentication (SF.I&A), used e.g. by TSF_Admin and TSF_Auth.
- Secure management of TOE resources (SF.SecureManagement)
- PIN management (SF.PIN)
- Transaction management (SF.Transaction)

The TOE Security Functions marked in gray are not used within the TOE of this security target. In the list above, examples are given for the use of the platform TSF by the TOE of this security target; a detailed analysis is not in the scope of this security target.

7.2. Mapping of TOE Security Functional Requirements and TOE Security Functions

Each TOE security functional requirement is implemented by at least one security function. The mapping of TOE Security Requirements and TOE Security Functions is given in the following table. If iterations of a TOE security requirement are covered by the same TOE security function the mapping will appear only once. The description of the TSF is given in section 7.1.

	TSF_Access	TSF_Admin	TSF_Secret	TSF_Crypto	TSF_SecureMessaging	TSF_Auth	TSF_Javacard
FCS_COP.1/TDES				x			x
FCS_COP.1/MAC				x			x
FCS_CKM.4			x				x
FCS_RNG.1							x
FIA_UAU.1	x					x	
FIA_UAU.3	x					x	
FIA_UID.1	x						(x)
FDP_ITC.1/GP	x	x				x	x
FDP_ITC.1/ACT	x	x		x	x	x	(x)
FDP_UIT.1	x			x	x	x	(x)
FDP_ACC.1	x			x		x	(x)
FDP_ACF.1	x			x		x	(x)

FPT_EMSEC.1							x
FPT_FLS.1							x
FPT_PHP.3							x
FMT_SMF.1		x				x	(x)
FMT_SMR.1		x				x	(x)

Table 2: Mapping of TOE Security Functional Requirements and TOE Security Functions.

Please note that the characteristics and security functions of the platform (subsumed in TSF_Javacard) are necessary for the fulfillment of all requirements, since all other TOE security functions are implemented as Javacard code. This is indicated by a (x) in the according row.

References

Common Criteria

[CC_1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2006-09-001, Version 3.1, Revision 1, September 2006

[CC_2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2007-09-002, Version 3.1, Revision 2, September 2007

[CC_3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2007-09-003, Version 3.1, Revision 2, September 2007

[CC_4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2007-09-004, Version 3.1, Revision 2, September 2007

Protection Profiles

[PP0002] PP conformant to Smartcard IC Platform Protection Profile, Version 1.0, July 2001; registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0002-2001

[PP0035] Security IC Platform Protection Profile, Version 1.0, June 2007; registered and certified by BSI (Bundesamt für Sicherheit in der Informationstechnik) under the reference BSI-PP-0035-2007

[PP_Javacard] Java Card System - Minimal Configuration Protection Profile, Version 1.1, May 2006, part of: Java Card Protection Profile Collection, Version 1.1, May 2006

Cryptographic specifications

[FIPS46-3] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION FIPS PUB 46-3, DATA ENCRYPTION STANDARD (DES), Reaffirmed 1999 October 25, U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology

[FIPS197] Federal Information Processing Standards Publication 197, ADVANCED ENCRYPTION STANDARD (AES), U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, November 26, 2001

Other

[ISO7816-4] ISO 7816, Identification cards – Integrated circuit(s) cards with contacts, Part 4: Organization, security and commands for interchange, FDIS 2004

[ISO9797-1] ISO/IEC 9797-1:1999: Information technology - Security techniques – Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher

[ANSIX9.31] American National Standards Institute, ANSI X9.31-1998: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry (rDSA), 1998.

[AIS20] Anwendungshinweise und Interpretationen zum Schema, AIS 20: Funktionalitaetsklassen und Evaluationsmethodologie fuer deterministische Zufallszahlengeneratoren, Version 1, 02.12.1999, Bundesamt fuer Sicherheit in der Informationstechnik

[Java_RES] Runtime Environment Specification, Java Card(tm) Platform, Version 2.2.2, March 2006, Sun Microsystems

[ASE_JCOP081] Security Target Lite „NXP J3A081, J2A081 and J3A041 Secure Smart Card Controller Rev. 3“, Rev. 01.02; NXP, December 2010.

[ZertJCOP081] Certification Report BSI-DSZ-CC-0675-2011 for NXP J3A081, J2A081 and J3A041 Secure Smart Card Controller Revision 3 from NXP Semiconductors Germany GmbH; BSI, April 2011.

[ZertIC081] Certification Report BSI-DSZ-CC-0555-2009 for NXP Smart Card Controller P5CD081V1A and its major configurations P5CC081V1A, P5CN081V1A, P5CD041V1A, P5CD021V1A and P5CD016V1A each with IC dedicated Software from NXP Semiconductors Germany GmbH Business Line Identification; BSI, November 2009.

[Guidance1] Operational user guidance manuel for LEGIC AFS4096-JP12, LA-33-200c-en “Activation process for LEGIC all-in-one area”, LEGIC Identsystems Ltd., 2012

[Guidance2] Preparational user guidance for LEGIC AFS4096-JP12, LA-33-205c-en “Initialisation Process for LEGIC initialised Smart Cards”, LEGIC Identsystems Ltd., 2012

[Guidance3] User Manual, LA-23-616a-en “LEGIC card-in-card AFS4096-JP12 V1.2”, LEGIC Identsystems Ltd., 2012

Glossary

AES	The AES (Advanced Encryption Standard) has been defined as a standard for symmetric data encryption. It is a block cipher with a block length of 128 bit and key lengths of 128, 192 and 256 bit.
Authentication	Authentication defines a procedure that verifies the identity of the communication partner. The most elegant method is based on the use of so called digital signatures.
Block cipher	An algorithm processing the plaintext in bit groups (blocks). Its alternative is called stream cipher.
Cryptography	In the classical sense, the science of encrypting messages. Today, this notion comprises a larger field and also includes problems like authentication or digital signatures.
DES	Data Encryption Standard. Symmetric 64 bit block cipher, which was developed (first under the name Lucifer) by IBM. The key length is 64 bit of which 8 bit serve for a parity check. DES is the classic among the encryption algorithms, which nevertheless is no longer secure due to its insufficient key length. Alternatives are Triple-DES or the successor AES.
Hash function	A function which forms the fixed-size result (the hash value) from an arbitrary amount of data (which is the input). These functions are used to generate the electronic equivalent of a fingerprint. The significant factor is that it must be impossible to generate two entries which lead to the same hash value (so called collisions) or even to generate a matching message for a defined hash value. Common hash functions are RIPEMD-160 and SHA-1, each having hash values with a length of 160 bit as well as the MD5, which is still often used today having a hash value length of 128 bit.
Integrity	The test on the integrity of data is carried out by checking messages for changes during the transmission by the receiver. Common test procedures employ Hashfunctions, MACs (Message Authentication Codes) or – with additional functionality – digital signatures.
Javacard	A smart card with a Javacard operation system. In the context of this security target, Javacard refers to the platform defined in section 1.1.
MAC	Algorithm that expands the message by means of a secret key by special redundant pieces of information, which are stored or transmitted together with the message. To prevent an attacker from targeted modification of the attached redundancy, requires its protection in a suitable way.
Passphrase	A long, but memorable character sequence (e.g. short sentences with punctuation) which should replace passwords as they offer more security.
Pseudo random number	Many cryptographic mechanisms require random numbers (e.g. in key generation). The problem, however, is that it is difficult to implement true random numbers in software. Therefore, so called pseudo-random number generators are used, which then should be initialized with a real random element (the so called *seed*).
Random numbers	Many cryptographic algorithms or protocols require a random element, mostly in form of a random number, which is newly generated in each case. In these cases, the security of the procedure depends in part on the suitability of these random numbers. As the generation of real random numbers within computers still imposes a problem (a source for real random events can in fact only be gained by exact observation of physical events, which is not easy to realize for a software), so called pseudo random numbers are used instead.

Secure messaging	Secure messaging using encryption and message authentication code according to ISO/IEC 7816-4.
SFR	Security functional requirement.
Skimming	Imitation of a reader system to read the data fields or parts of it via the contactless communication channel of the TOE.
Smart card	A smart card is a chip card which contains an internal micro controller with CPU, volatile (RAM) and non-volatile (ROM, EEPROM) memory, i.e. which can carry out its own calculations in contrast to a simple storage card. Sometimes a smart card has a numerical coprocessor (NPU) to execute public key algorithms efficiently. Smart cards have all of their functionality comprised on a single chip (in contrast to chip cards, which contain several chips wired to each other). Therefore, such a smart card is ideal for use in cryptography as it is almost impossible to manipulate its internal processes.
Stream cipher	Symmetric encryption algorithm which processes the plaintext bit-by-bit or byte-by-byte. The other usually employed class of procedures comprises so called block cipher.
Symmetric cipher	Encryption procedure using the same key for enciphering and deciphering (or, in which these two keys can simply be derived from each other). One distinguishes between block ciphers processing plaintext in blocks of fixed length (mostly 64 or 128 bit) and stream ciphers working on the basis of single characters.
TOE	Target of evaluation.
User data	The TOE provides a memory area which can be accessed by authenticated terminals. The data stored in this memory area by authenticated terminals are denoted “user data”. The memory area provided by the TOE may be divided into different smaller memory areas which are used by different applications; the management of this application-specific substructure is solely controlled by the authenticated terminal and not by the TOE itself.