COMMON CRITERIA PROTECTION PROFILE

SECURITY IC PLATFORM

Draft Version 1.1

TURKISH STANDARDS INSTITUTION

0. REVISION HISTORY

Revision No:	Revision	Date
Draft 1.0	First Draft	26 August 2013
Draft 1.1	Response to Observation No:1	23 November 2014

TABLE OF CONTENTS

FIGURES LIST

TABLES LIST

1. INTRODUCTION

1.1 PP Reference

1.1.1 TITLE

Security IC Platform

1.1.2 VERSION

Draft v1.1

1.1.3 AUTHOR

Turkish Standards Institution

1.1.4 PUBLICATION DATE

23.11.2014.

1.2 TOE OVERVIEW

1.2.1 TOE TYPE

The TOE is the Security IC, which constitutes the execution environment for smart card applications.

The packaging, external components such as battery and antenna, physical card are out of scope of the PP. The activities such as, packaging, composite product manufacturing, are also taken out of evaluation context.

1.2.2 TOE USAGE AND MAJOR SECURITY PROPERTIES

The TOE is used as a platform for following types of security sensitive applications:

- Identification and Authentication
- Electronic Fare and Purse
- Electronic Signature
- Secure Data Storage

The TOE ensures the security of the application data and the embedded operating system from attackers in terms of confidentiality and integrity; and ensures correct operation of embedded operating system. The TOE also provides to the embedded operating system the security service: "Random Number Generation".

The threats that the TOE counters, the needs of the embedded operating system are given in the Security Problem Definition of this PP. But before proceeding and going in the details, the objective of the TOE is to provide the applications and consumers a secure platform resilient to the attacks targeting the platform.

The current PP addresses two different configurations of the TOE, which are: the ROM based and Flash based. The difference arises from the lifecycle changes and the flexibility that the Flash based technology offers brings additional security concerns.

The current Protection Profile defines the security problem, the security objectives for the TOE, the security objectives for the environment, and the security requirements. The security problem defines the external entities that interact with the TOE, the assets and the security services, the threats, the assumptions related with environment and the organisational security policies that the TOE must comply with. The environment consists of manufacturing site, end-user environment and card issuer environment. From the TOE's point of view the difference between the card issuer and the end-user environment exists if and only if the TOE is flash based and flash loader is not disabled before delivery from the manufacturer to the card issuer. Security objectives for the environment give out what are expected from the environment in order to TOE be secure. There is nothing expected from the end-user environment, it is all assumed as hostile environment due to the nature of smart card applications. But there exist expectations from manufacturing and card issuer sites. Finally in the PP, both the security functional requirements and security assurance requirements are given. The rationale describing how security objectives solve the security problem and the rationale how security requirements fulfil the TOE security objectives are also given.

1.2.3 NON-TOE HARDWARE/FIRMWARE/SOFTWARE

None

1.3 TOE DEFINITION

1.3.1 TOE CONFIGURATIONS

There are two different TOE configurations that depend on the choice of embedded OS storage technology exist in this PP:

- The ROM Based
- The Flash Based

ROM Storage: This configuration requires the Embedded OS to be delivered prior to the manufacturing of the TOE. Embedded OS is stored in the ROM Memory of the TOE and rewrite is not possible.

Flash Storage: Flash Technology enables the embedded OS to be loaded after the TOE is manufactured. There is Flash Loader Software in the TOE within this configuration and provides functionality to load the embedded OS to the Flash Memory. There are several options to load the embedded OS; the manufacturer may load the embedded OS and disable flash loader prior to the delivery to the Card Issuer; or the TOE is delivered without ES and flash loader is active.

Starting from the lifecycles of the different configurations, differences for two configurations exist within this PP. Therefore any TOE that claims conformance to this PP has to state the embedded OS storage technology it supports in the PP conformance statement, whether ***ROM based*** or ***Flash Based***.

The TOE configuration differs in the way that how the embedded operating system is stored or loaded. Embedded operating system can be embedded during the manufacturing by using the Hard ROM masks which are prepared prior to the TOE manufacturing or may be loaded afterwards the manufacturing by a flash loader program if TOE is based on Flash Technology. Usage of Flash or ROM technology changes the lifecycle of the TOE and due to the flexibility the Flash Technology brings there are additional security concerns for the Flash based TOE's. For a TOE that is claiming conformance to this PP, which technology this TOE supports needs to be given in its security target.

Besides the configurations stated in this PP; additional threats, organisational security policies, TOE security objectives and security requirements can be added. Any other addition to the assumptions and environment security objectives requires consistency to other elements of the PP.

Note about the Configurations:

ROM based technology is simpler and the only difference between the ROM based and Flash based TOE is that Flash based TOE's have additional security items. Therefore first the SPD, SOs and SRs of ROM based technology are given afterwards if exist additional items for Flash based TOE are given.

1.3.2 THE TOE CONTENTS

The TOE consists of:

- Hardware of Security IC
- Security IC Dedicated Software
- IC Unique Identification Data + Configuration Data (optional)
- Security Guidance Documentation

The Hardware of Security IC: The hardware consists of CPU, volatile and non-volatile memories, I/O components and security components.

Security IC Dedicated Software: Any software other than the embedded OS is the Security IC Dedicated Software. Security IC Dedicated Software consists of IC Dedicated Test Software, IC Dedicated Support Software and if embedded OS is stored on Flash Memory additionally Flash Loader Software.

Definition of IC Dedicated Test Software and IC Dedicated Support Software is given in the PP0035 [1][1] as:

. *'The "IC Dedicated Test Software" is not usable after TOE Delivery. Therefore, this software (or parts of it) is seen only as a "test tool" though being delivered as part of the TOE. The IC Dedicated Test Software does not provide security functionality after TOE Delivery and is only used to support testing of the TOE during production. However, it must be verified that it cannot be abused after TOE Delivery: this is evaluated according to the Common Criteria assurance family AVA_VAN.*

. *In contrast, the "IC Dedicated Support Software" does provide functions after TOE Delivery. Therefore, during evaluation it is treated as all other parts of the TOE. The IC Dedicated Support Software may be stored in the ROM or may be delivered as source code or libraries in addition to the hardware. '*

IC Unique Identification Data: Unique identification is the requirement of the card issuer. Card Issuer, requires that each IC has unique identification number to track the ICs and to develop an authentication mechanism that will produce different authentication reference data for each IC.

Configuration Data (optional)**:** IC manufacturers offer their consumers wide range of products with different sizes of memories and with different functionalities. They usually use the same physical product with different configurations to provide the range of products. Different configurations are implemented with mechanisms to disable the functionalities and with some configuration data that limits or blocks the memory size and/or functionalities. A vendor may not offer different configurations of same physical product, in this case configuration data does not exit, so it is optional.

[1] PP0035: Security IC Platform Protection Profile, Version 1.0, 15.06.2007 (Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035).

Security Guidance Documentation: Contains the information about the secure usage of the IC. The Security Guidance Documentation is also evaluated and if and only if a TOE is used accordingly to this security guidance it is considered as certified. Any failure in compliance to the security guidance will break the security certificate and the TOE can be assumed as in the evaluated configuration state.

1.3.3 THE TOE PHYSICAL VIEW

The Physical Hardware consists of:

- CPU
- Volatile and Non-Volatile Memories
- Security Components[2]
- Communication Interfaces

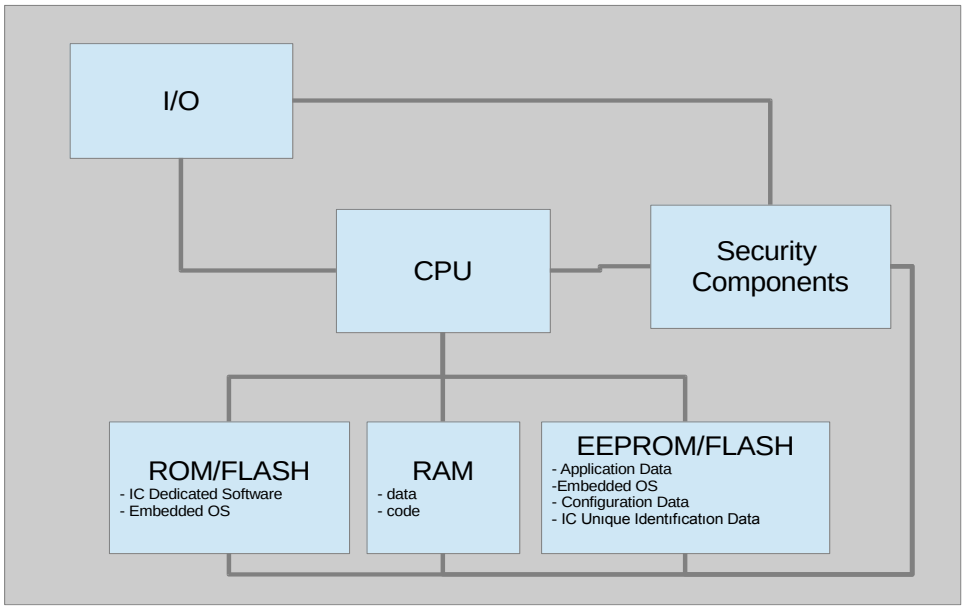


Figure 1: Secure IC Diagram

Data Stored on the Non-Volatile Memories

- IC Dedicated Software
- Embedded OS
- Application Data
- Configuration Data
- IC Unique Identification Data

Separately Delivered Item(s):

[2] Security Components cover all the mechanisms for protection of the TOE such as sensors, CPU comparator, integrity checkers, and shields.

- Guidance Documentation

1.3.4 THE TOE LOGICAL VIEW

The logical view of the TOE is in the below figure 2. The IC Dedicated Test Software, IC Dedicated Support Software and Security IC Hardware together form the TOE and set the boundaries of it. In the operational phase the TOE interacts only with the embedded OS. Before delivery, the TOE manufacturer interacts with the IC Dedicated Test Software and IC Dedicated Support Software.

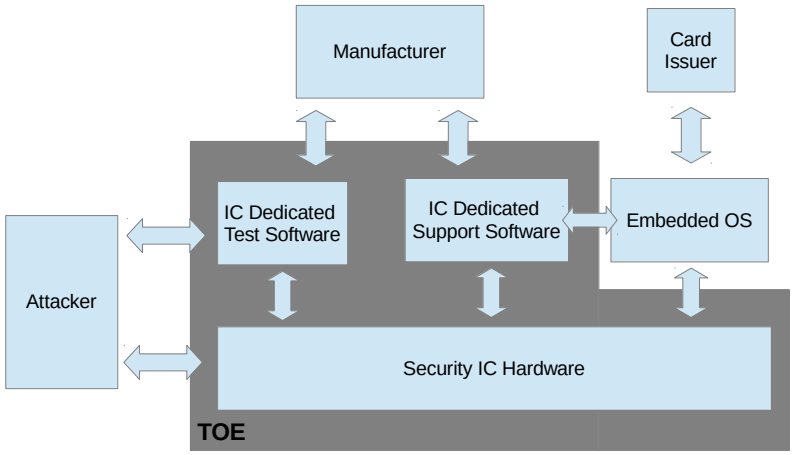


Figure 2: Logical View of the TOE for ROM based configuration

1.3.4.1 THE LOGICAL VIEW OF THE FLASH BASED TOE

For the Flash based TOE, the Flash Loader software additionally takes place in the TOE. The manufacturer or the card issuer uses the Flash Loader to load the embedded operating system to the TOE and before delivery they unlock the flash loader to be further used. Attacker may also interact with the flash loader to abuse the functionality it offers, to load its own embedded operating system. Below the logical diagram for the flash based TOE is given.

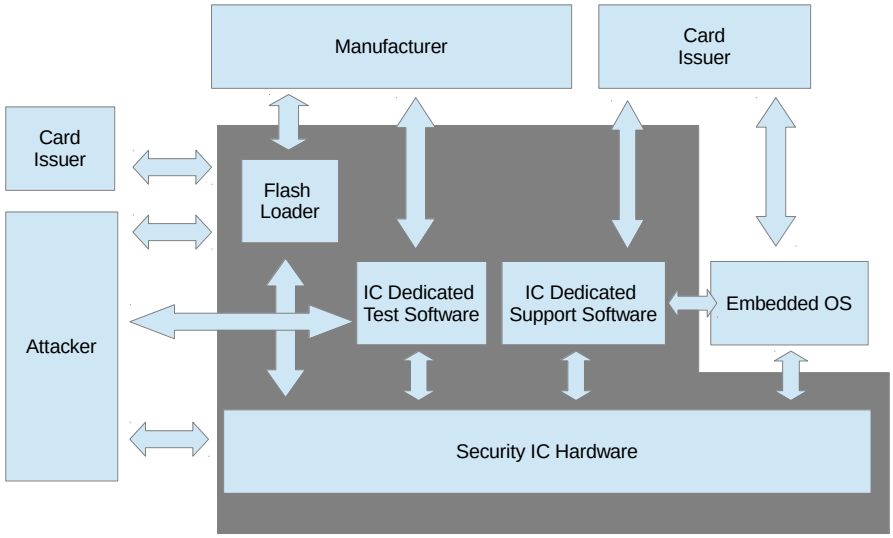


Figure 3: Logical Model for the Flash based TOE

1.3.5 THE TOE LIFE-CYCLE MODEL

1.3.5.1 PHASES OF THE LIFE-CYCLE

Following phases exist within life-cycle of the TOE*,

- Design and Development Phase
- Manufacturing Phase
- Card Issuer Phase

*Packaging and composite product manufacturing is taken out since they implement out of scope physical characteristics of the security IC.

Design and Development Phase:

In this phase, design of the TOE and the development of the IC Dedicated Software (for flash based case also flash loader) are performed. Design data and software is delivered to the manufacturing phase. Also the guidance documentation is delivered to the embedded operating system developer.

Manufacturing Phase:

During manufacturing phase, for the ROM based case embedded operating system is received from the Embedded OS Developer and the TOE is

- manufactured (for the ROM based case also the embedded operating system is put into);
- tested (faulty ones are detected and removed)
- test functionality is disabled
- embedded operating system is loaded (for the flash based case and if embedded operating system is loaded by the manufacturer, if embedded operating is loaded by the card issuer this step is omitted)

- flash loader is disabled (for the flash based case and if embedded operating system is loaded by the manufacturer, if embedded operating is loaded by the card issuer this step is omitted)
- IC Unique Identification is written
- the TOE is delivered to the Card Issuer

Card Issuer Phase:

- the TOE is received from manufacturer (the TOE comes with the embedded operating system for ROM based case and for flash based case if embedded operating system is loaded to the TOE by the manufacturer)
- the Card Issuer loads the embedded operating system to the TOE (for the case Card Issuer loads the embedded operating system) and disables the flash loader
- Now the only user of the TOE is the embedded operating system and attacker might try the security policy of the card issuer

1.3.5.2 MANUFACTURER LOADS THE ES LIFE-CYCLE MODEL

This life-cycle happens when ES Developer puts the ES to the TOE. From life-cycle point of view, usage of the Flash or ROM technology does not make any difference, since both cases result with the TOE delivered to Card Issuer with ES loaded and ES reloading is disabled.

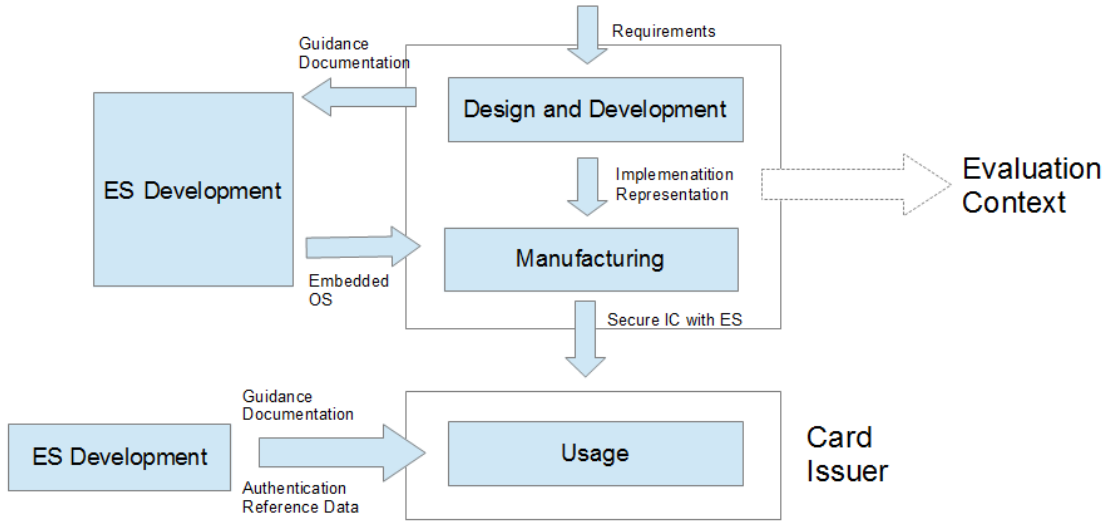


Figure 4: Life-cycle Manufacturer loads the ES

Authentication Reference Data: That is the data used by the card issuer to authenticate itself to the TOE for ES loading function.

1.3.5.3 CARD ISSUER LOADS THE ES LIFE-CYCLE MODEL

In this case the TOE is received from the manufacturer without the ES loaded and with flash loader functional. It is the card issuer responsibility to load the embedded operating system and to disable the flash loader after loading the embedded operating system.

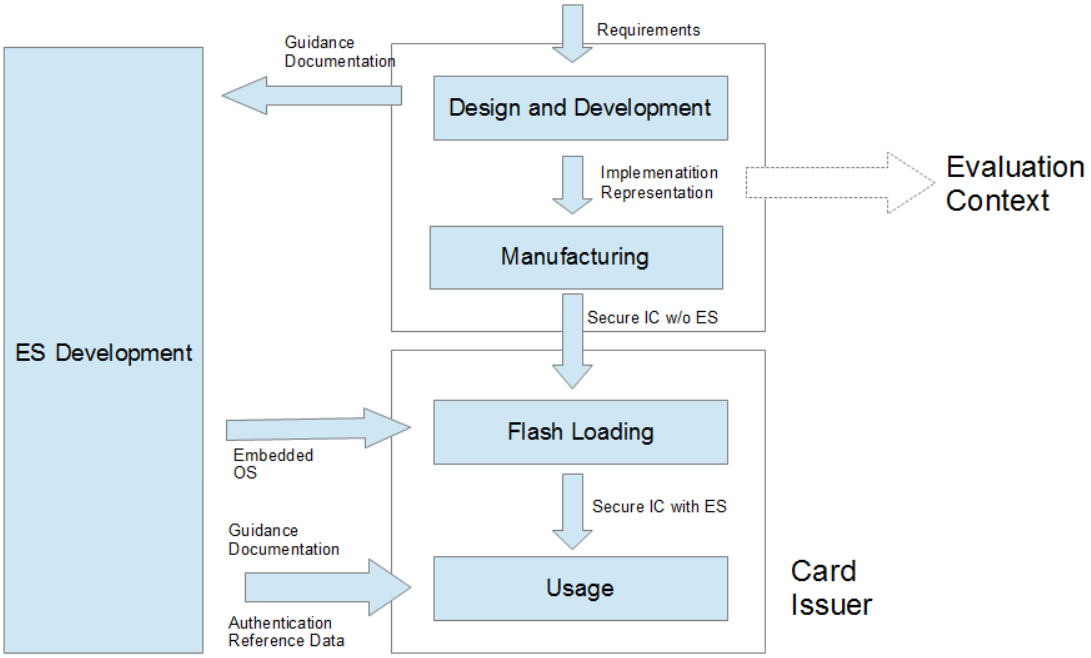


Figure 5: : Life-cycle Card Issuer loads the ES

Authentication Reference Data: That is the data used by the card issuer to authenticate itself to the TOE for ES loading function.

2. CONFORMANCE CLAIMS

2.1 CC CONFORMANCE CLAIM

This Protection Profile claims conformance to the

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

The

- Common Methodology for Information Technology Security Evaluation (CEM), Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012

has to be taken into account

2.2 PP CLAIM

This PP does not claim conformance to any other PP.

2.3 PACKAGE CLAIM

This PP is conforming to assurance package EAL5 augmented with ALC_DVS.2 and AVA_VAN.5 defined in CC Part 3.

2.4 CONFORMANCE STATEMENT

The Protection Profile requires **strict** conformance for the ST or PP claiming conformance to this PP.

3. SECURITY PROBLEM DEFINITION

3.1 INTRODUCTION

3.1.1 ASSETS AND SECURITY SERVICES

The assets to be protected and the security services to be given and be protected are: **Application Data** (asset) and **Random Number Generation** (security service).

Application Data: Application data is the all data that belongs to ES and managed by ES. Application data can be used by the ES as TSF Data or User Data that depends on the usage context of the ES and Security IC might not distinguish the difference between so all are accepted as application data by the TOE.

Random Number Generation: The TOE provides to the users of it (here only the ES), the random number. Random numbers are essential for the security applications that ES serves for.

Besides the assets and the security services the TSF Data needs to be protected by the TOE which are **IC Unique Identification Data** and **the ES**. Unique identification of the ICs, that are manufactured and delivered to the Card Issuer, is a requirement that results from the organizational security policy: O.Unique_Identification. ES stored on non-volatile memory also needs to be protected from disclosure or manipulation. Manipulation of ES results to incorrect functioning of the ES, which is a violation of one of the main security concerns of the TOE. Disclosure of ES gives opportunities to the attackers to reverse engineer ES and discover any weaknesses of it.

Additional Items:

Any security service and TSF data belonging to that security service may be added by ST author.

3.1.2 SUBJECTS AND EXTERNAL ENTITIES

Manufacturer: Manufacturer performs design, development and manufacturing of the TOE. Manufacturer also performs following administrative actions: disabling test functions, writing the IC unique identification Data, loading ES and disabling Flash Loader. Loading ES and disabling Flash Loader is only performed by the manufacturer if the TOE is flash based and ES loading is performed by manufacturer. For ROM based these activities do not exist and for card issuer loads the ES case these activities are performed by Card Issuer.

ES: ES is the only user of the TOE. The TOE serves to the ES by providing a secure execution environment and secure storage of its data. The TOE also provides unique identification of itself and random numbers.

Card Issuer: Card issuer is the legitimate owner of the TOE. It normally uses the TOE via embedded OS. Card Issuer trusts the TOE that the TOE provides secure execution environment and secure storage to ES. If the TOE is flash based and ES is loaded by the Card Issuer then it performs administrative actions: loading the ES and disabling the Flash Loader.

Attacker: The attacker is the entity who tries to undermine the security policies of the card issuer. Attacker is assumed to possess *high attack potential*.

3.2 THREATS

PROBING AND MANIPULATION THREATS:

T.Probing_on_Data_Storage: An attacker may physically attack to the memories of the TOE to gain illicit access to the application data.

T.Probing_on_Data_Transfer: An attacker may probe the internals of the TOE to gain illicit access to the application data.

T.Manipulation_on_Data_Storage: An attacker may physically attack to the memories of the TOE to make changes to the application data.

T.Manipulation_on_Data_Transfer: An attacker may physically attack to the internals of the TOE to make changes to the application that is transferred between internal parts of the TOE.

T.Manipulation_on_Execution: An attacker may physically attack to the CPU operations to alter the execution of the ES.

T.Manipulation_on_RND: An attacker may physically attack to the random number entropy source to cause the TOE to generate random numbers with insufficient quality.

LEAKAGE AND EMISSION THREATS

T.Information_Leakage_Surface: An attacker may monitor and interpret the emissions emanated from the physical surface of the TOE to disclose the application data.

T.Information_Leakage_Contacts: An attacker may monitor the power consumption, timing of operation and other observables to interpret and get access to the application data.

ENVIRONMENTAL STRESS THREATS

T.Environmental_Stress_Application: Attacker may apply temperature, frequency, voltage outside of the standard operating conditions to force the TOE to malfunction.

FUNCTIONALITY ABUSE THREATS

T.Abuse_of_the_Test_Functions: Attacker may try to use the test functions to gain illicit access to the application data.

THREAT FOR THE FLASH LOADER CONFIGURATION

T.Abuse_of_the_Flash_Loader: Attacker may try to load an unauthentic ES to the ES by using the functionality of the Flash Loader. ***(Applies to the Flash based TOEs)***

3.4 ORGANISATIONAL SECURITY POLICIES

P.IC_Unique_Identification_Data: The TOE should be uniquely identified by ES.

P.Random_Number_Generation: The TOE should provide random numbers to the ES.

ORGANISATIONAL SECURITY POLICY FOR THE FLASH LOADER CONFIGURATION

P.Embedded_OS_Load: The TOE should provide ES loading capability to the Card Issuer or the Manufacturer.

3.5 ASSUMPTIONS

A.Guidance_Documents_Compliance: It is assumed that the ES complies with the security guidance of the TOE. The security of the TOE depends on the ES, and there may be security mechanisms which are affective if and only if the ES correctly uses them.

4 SECURITY OBJECTIVES

4.1 SECURITY OBJECTIVES FOR THE TOE

OT.Data_Access_Control: The TOE must protect the data stored and processed on the TOE from unauthorized access through physical probing. Access to the memories should only be granted to the ES. Unauthorized access covers the physical probing which may be performed by an attacker.

OT.Data_Integrity: The TOE must detect the manipulation of the data stored on the TOE.

OT.Internal_TOE_Transfer_Confidentiality_Protection: The TOE must protect the confidentiality of the data transferred between internal parts of the TOE from probing.

OT.Internal_TOE_Transfer_Integrity_Protection: The TOE must protect the integrity of the transferred data between internal parts of the TOE.

OT.CPU_Operation_Protection: The TOE must have functionality to protect against any manipulation to embedded OS execution.

OT.Random_Number_Generation_Protection: The TOE must have functionality to protect the number random number generation functionality.

OT.Environmental_Stress_Protection: The TOE must have mechanism(s) to protect itself from environmental stress.

OT.Side_Channel_Protection: The TOE must have protection against T.Information_Leakage_Contacts. The TOE should not emit any useful information that might help the attacker to determine or guess the confidential data that is processed by the TOE. .

OT.Test_Functions_Disable_Mechanism: The TOE must have mechanism allowing the manufacturer to irreversibly disable the test functionality.

OT.Random_Number_Generation: The TOE should have mechanisms to provide random numbers to the embedded operating system.

OT.Unique_ID_Storage: The TOE must have functionality to store Unique Identification Data. This data should only be writable by the manufacturer.

ADDITIONAL OBJECTIVES FOR THE FLASH LOADER CONFIGURATION

OT.Flash_Loader_Functionality: The TOE must have embedded operating system loading functionality.

OT.Flash_Loader_Authorization: The TOE must allow only authenticated entities to use the flash loader.

OT.Flash_Loader_Disable_Mechanism: The TOE must have mechanism allowing the manufacturer and/or the card issuer to irreversibly disable the flash loader functionality.

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

OE.Test_Functions_Disable: The manufacturer must ensure that for every TOE before delivery, test functionality has been disabled.

OE.Unique_Identification: The manufacturer must use the unique ID storage mechanism of the TOE properly. The manufacturer must ensure that IDs written to the TOEs are unique.

OE.Embedded_OS: The embedded operating system must comply with the security guidance.

ADDITIONAL OBJECTIVES FOR THE FLASH LOADER CONFIGURATION

OE.Flash_Loader_Disable: Depending on the life-cycle of the TOE, whether the manufacturer or the card issuer must ensure that any TOE that will be delivered, the flash loader functionality had been disabled.

4.3 SECURITY OBJECTIVES RATIONALE

4.3.1 SECURITY OBJECTIVES RATIONALE TABLE

The following table provides an overview for security objectives coverage.

No	SPD	SO
1.	T.Probing_on_Data_Storage	OT.Data_Access_Control
2.	T.Probing_on_Data_Transfer	OT.Internal_TOE_Transfer_Confidentiality_Protection
3.	T.Manipulation_on_Data_Storage	OT.Data_Access_Control, OT.Data_Integrity
4.	T.Manipulation_on_Data_Transfer	OT.Internal_TOE_Transfer_Confidentiality_Protection, OT.Internal_TOE_Transfer_Integrity_Protection
5.	T.Manipulation_on_Execution	OT.CPU_Operation_Protection
6.	T.Manipulation_on_RND	OT.Random_Number_Generation_Protection
7.	T.Information_Leakage_Surface	OT.Internal_TOE_Transfer_Confidentiality_Protection
8.	T.Information_Leakage_Contacts	OT.Side_Channel_Protection
9.	T.Environmental_Stress_Application	OT.Environmental_Stress_Protection
10.	T.Abuse_of_the_Test_Functions	OT.Test_Functions_Disable_Mechanism, OE.Test_Functions_Disable
11.	P.IC_Unique_Identification_Data	OT.Unique_ID_Storage, OE.Unique_Identification
12.	P.Random_Number_Generation	OT.Random_Number_Generation
13.	A.Guidance_Documents_Compliance	OE.Embedded_OS

Table 1: Security Objectives Rationale

ADDITIONAL RATIONALE FOR THE FLASH LOADER CONFIGURATION

14.	P.Embedded_OS_Load	OT.Flash_Loader_Functionality
15.	T.Abuse_of_the_Flash_Loader	OT.Flash_Loader_Authorization, OT.Flash_Loader_Disable_Mechanism,

		OE.Flash_Loader_Disable

Table 2:Additional Rationale for the Flash Loader

4.3.2 SECURITY PROBLEM JUSTIFICATION

In this section justification of security problem by security objectives is given.

- **T.Probing_On_Data_Storage**

OT.Data_Access_Control protects the data stored and processed on the TOE from physical probing; and so the objective OT.Data_Access_Control counters the threat T.Probing_On_Data_Storage.

- **T.Probing_on_Data_Transfer**

OT.Internal_TOE_Transfer_Confidentiality_Protection protects the confidentiality of the transferred data between internals of the TOE; and so it covers the threat T.Probing_on_Data_Transfer.

- **T.Manipulation_on_Data_Storage**

OT.Data_Integrity protects the integrity of the data stored and processed on the TOE from physical manipulation. And the OT.Data_Access_Control protects against reading the data, prevents the attacker knowing the physical location and content of the data. So with OT.Data_Access_Control, even if OT.Data_Integrity does not exist; an attacker may manipulate the data and can not foresee the changes he or she makes. OT.Data_Integrity and OT.Data_Access_Control covers this threat.

- **T.Manipulation_on_Data_Transfer**

OT.Internal_TOE_Transfer_Integrity_Protection protects the integrity of the data transferred between internal parts of the TOE from physical manipulation. And the OT.Internal_TOE_Transfer_Confidentiality_Protection prevents the attacker from compromising the data in transfer so making reasonable changes is not possible. So with OT.Data_Access_Control, even if OT.Data_Integrity does not exist; an attacker may manipulate the data and can not foresee the changes he or she makes. OT.Internal_TOE_Transfer_Confidentiality_Protection and OT.Internal_TOE_Transfer_Integrity_Protection covers this threat.

- **T.Manipulation_on_Execution**

OT.CPU_Operation_Protection protects the CPU operations from manipulation. So the objective OT.CPU_Operation_Protection covers T.Manipulation_on_Execution.

- **T.Manipulation_on_RND**

OT.Random_Number_Generation_Protection protects the Random Number Generation functionality from manipulation. So the objective OT.Random_Number_Generation_Protection covers T.Manipulation_on_RND.

- **T.Information_Leakage_Surface**

OT.Internal_TOE_Transfer_Confidentiality_Protection, protects the confidentiality of the transferred data, so interpretation of the emissions is not possible. OT.Internal_TOE_Transfer_Confidentiality_Protection covers this threat.

- **T.Information_Leakage_Contacts**

OT.Side_Channel_Protection covers the threat T.Information_Leakage_Contacts.

- **T.Environmental_Stress_Application**

OT.Environmental_Stress_Protection covers the T.Environmental_Stress_Application

- **T.Abuse_of_the_Test_Functions**

OT.Test_Functions_Disable_Mechanism and OE.Test_Functions_Disable cover the T.Abuse_of_the_Test_Functions.

- **T.Abuse_of_the_Flash_Loader**

OT.Flash_Loader_Authorization prevents the usage of Flash Loader by unauthorized entities, so even if during delivery or within the card issuer facility attacker accesses to the TOE, he or she will not be able to use the flash loader to load an unauthentic embedded operating system. OT.Flash_Loader_Disable_Mechanism enables the manufacturer or the card issuer to disable the flash loader functionality irreversebly. OE.Flash_Loader_Disable declares that the manufacturer or the card issuer (depending on the life-cycle model) to disable the flash loader before delivery from their facility. So the objectives OT.Flash_Loader_Authorization, OT.Flash_Loader_Disable_Mechanism, and OE.Flash_Loader_Disable together covers this threat.

- **P.IC_Unique_Identification_Data**

OT.Unique_ID_Storage enables the TOE to have ID storage capability and OE.Unique_Identification requires that manufacturer creates a unique identification data for the TOE and writes it to the TOE before delivery to the card issuer. So OT.Unique_ID_Storage and OE.Unique_Identification covers the P.IC_Unique_Identification_Data

- **P.Random_Number_Generation**

OT.Random_Number_Generation covers the policy P.Random_Number_Generation.

- **P.Embedded_OS_Load**

OT.Flash_Loader_Functionality covers the policy P.Embedded_OS_Load.

- **A.Guidance_Documents_Compliance**

A.Guidance_Documents_Compliance assumes that the embedded operating system complies with the security guidance documentation, OE.Embedded_OS fulfills this assumption.

5. EXTENDED COMPONENTS

Following components are added:

- FCS_RND.1
- FPT_SCP.1

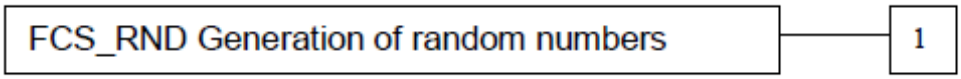
5.1 CLASS FCS CRYPTOGRAPHIC SUPPORT

FAMILY FCS_RND 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_RND.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RND.1

There are no management activities foreseen.

Audit: FCS_RND.1

There are no actions defined to be auditable.

FCS_RND.1 **Quality metric for random numbers**

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: a defined quality metric].

5.2 CLASS FPT PROTECTION OF THE TSF

FAMILY FPT_SCP SIDE CHANNEL PROTECTION

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 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.

The family “Side Channel Protection (FPT_SCP)” is specified as follows.

Family behavior:

This family defines requirements to mitigate information leakage through time and power analysis.

Component leveling:

FPT_SCP.1 Side channel protection requires to not emit interface emanation enabling access to TSF data or user data.

Management: FPT_SCP.1

There are no management activities foreseen.

Audit: FPT_SCP.1

There are no actions defined to be auditable.

FPT_SCP.1 Side Channel Protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SCP.1 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

6.1.1 DATA ACCESS CONTROL AND TRANSFER PROTECTION

Probing threat can be modelled as the attacker tries to access to the physical memories and/or tries to monitor data traffic between the internal parts of the TOE. The security objectives "Data Access Control", and "Internal TOE Transfer Confidentiality Protection" states that the TOE should provide protection against this kind of threat. These objectives can be achieved by the enforcement of the ***Internal Data Access Policy*** which briefly states: "The TSF shall allow only the Embedded OS (the only legitimate user) to access all data stored on memories and transmitted between internal parts of the TOE".

Internal Data Access Policy is enforced by the three SFRs: FDP_ACC.2, FDP_ACF.1 and FDP_ITT.1. Both FDP_ACF.1 and FDP_ITT.1 are performing the denial of access to the attacker by encrypting the data and allowing access to the embedded operating system by decrypting it. So they depend on the cryptographic operations (key generation, encryption and decryption, key destruction). FCS_CKM.1, FCS_CKM.4 and FCS_COP.1 is also added to the Protection against Probing SFRs. FPT_SCP.1 protects the cryptographic operations from side channel attacks.

FDP_ACC.2 Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

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

FDP_ACC.2.1 The TSF shall enforce the *Internal Data Access Policy*[3] on *Attacker, Embedded OS and Application Data*[4] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control fulfilled by FDP_ACC.2

FMT_MSA.3 Static attribute initialisation not fulfilled but justified

FDP_ACF.1.1 The TSF shall enforce the *Internal Data Access Policy*[5] to objects based on the following: *Attacker, Embedded OS and Application Data, type of user*][6].

[3] [assignment: access control SFP]
[4] [assignment: list of subjects and objects]
[5] [assignment: access control SFP]
[6] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

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

- *If the subject is Embedded OS access is allowed*
- *If the subject is Attacker access is not allowed*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[7].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[8].

FDP_ITT.1 Basic internal transfer protection

Hierarchical to: No other components.

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

FDP_ITT.1.1 The TSF shall enforce the *Internal Data Access Policy*[9] to prevent the *disclosure, and modification*[10] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.1 also prevents any confidential data leak through emanations through the physical surface of the IC.

6.1.1.1 CRYPTOGRAPHIC SUPPORT TO DATA ACCESS CONTROL AND TRANSFER PROTECTION

FCS_CKM.1/SP Cryptographic key generation – Storage Protection

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation] fulfilled by FCS_COP.1/SP

FCS_CKM.4 Cryptographic key destruction fulfilled by FCS_CKM.4

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm][11] and specified

[7] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[8] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].
[9] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[10] [selection: disclosure, modification, loss of use]
[11] [assignment: cryptographic key generation algorithm]

cryptographic key sizes [assignment: cryptographic key sizes][12] that meet the following: [assignment: list of standards][13].

FCS_CKM.1/TP Cryptographic key generation – Transfer Protection

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation] fulfilled by FCS_COP.1/TP

FCS_CKM.4 Cryptographic key destruction fulfilled by FCS_CKM.4

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm][14] and specified cryptographic key sizes [assignment: cryptographic key sizes][15] that meet the following: [assignment: list of standards][16].

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] fulfilled by FCS_CKM.1

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method][17] that meets the following: [assignment: list of standards][18].

FCS_COP.1/SP Cryptographic operation – Storage Protection

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] fulfilled by FCS_CKM.1/SP

FCS_CKM.4 Cryptographic key destruction fulfilled by FCS_CKM.4

[12] [assignment: cryptographic key sizes]
[13] [assignment: list of standards]
[14] [assignment: cryptographic key generation algorithm]
[15] [assignment: cryptographic key sizes]
[16] [assignment: list of standards]
[17] [assignment: cryptographic key destruction method]
[18] [assignment: list of standards]

FCS_COP.1.1 The TSF shall perform *encryption and decryption* [19] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm][20] and cryptographic key sizes [assignment: cryptographic key sizes] [21] that meet the following: [assignment: list of standards][22].

FCS_COP.1/TP Cryptographic operation – Transfer Protection

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] fulfilled by FCS_CKM.1/TP

FCS_CKM.4 Cryptographic key destruction fulfilled by FCS_CKM.4

FCS_COP.1.1 The TSF shall perform *encryption and decryptio*[23] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm][24] and cryptographic key sizes [assignment: cryptographic key sizes][25] that meet the following: [assignment: list of standards][26].

FPT_SCP.1 Side Channel Protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SCP.1 The TSF shall ensure *attackers*[27] are unable to use the following interface *physical contacts*[28] to gain access to *Data Storage Protection keys, Data Transfer Protection keys*[29] and *Stored Data and Transferred Data*[30].

6.1.2 PROTECTION AGAINST MANIPULATION

The SFRS listed under the 6.1.2 Protection Against Manipulation protect Application Data, TSF Data and ES Execution, Random Number Generator from manipulation. FDI_SDI.2 protects the stored Application Data from

[19] [assignment: list of cryptographic operations]
[20] [assignment: cryptographic algorithm]
[21] [assignment: cryptographic key sizes]
[22] [assignment: list of standards]
[23] [assignment: list of cryptographic operations]
[24] [assignment: cryptographic algorithm]
[25] [assignment: cryptographic key sizes]
[26] [assignment: list of standards]
[27] [assignment: type of users]
[28] [assignment: type of connection]
[29] [assignment: list of types of TSF data]
[30] [assignment: list of types of user data]

manipulation. FDP_ITT.3 protects the internally transferred Application Data. And finally, FPT_TST.1 protects the ES Data, ES Execution and Random Number Generator.

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: integrity errors][31] on all objects, based on the following attributes: *for all application data*[32].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment: action to be taken][33].

FDP_ITT.3 Integrity monitoring

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control] fulfilled by FDP_ACC.1/PAP

FDP_ITT.1 Basic internal transfer protection fulfilled by FDP_ITT.1

FDP_ITT.3.1 The TSF shall enforce the *Internal Data Access Policy*[34] to monitor user data transmitted between physically-separated parts of the TOE for the following errors: [assignment: integrity errors][35].

FDP_ITT.3.2 Upon detection of a data integrity error, the TSF shall [assignment: specify the action to be taken upon integrity error][36].

FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment:

[31] [assignment: integrity errors]
[32] [assignment: user data attributes].
[33] [assignment: action to be taken]
[34] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[35] [assignment: integrity errors]
[36] [assignment: specify the action to be taken upon integrity error]

conditions under which self test should occur]][37] to demonstrate the correct operation of *CPU Operation and Random Number Generator*[38].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*[39].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of *TSF*[40].

6.1.3 ADDITIONAL PROTECTION AGAINST PHYSICAL ATTACKS

Memory and bus encryption and also error detection is not sufficient against physical attacks since still attacker may attack to these encryption and error detection mechanisms so additional physical protection is necessary.

First of all FPT_PHP.3 is necessary for the protection of the TSF.

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *physical probing and manipulation*[41] to the [assignment: list of TSF devices/elements][42] by responding automatically such that the SFRs are always enforced.

After deployment of FPT_PHP.3, FPT_PHP.2 requires the TOE to notify Embedded OS from physical attacks.

FPT_PHP.2 Notification of physical attack

Hierarchical to: FPT_PHP.1 Passive detection of physical attack

Dependencies: FMT_MOF.1 Management of security functions behaviour not fulfilled but justified

FPT_PHP.2.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.2.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FPT_PHP.2.3 For [assignment: list of TSF devices/elements for which active detection is required][43], the TSF shall monitor the devices and elements and notify *Embedded OS*[44] when physical tampering with the TSF's devices or TSF's elements has occurred.

[37] [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]]
[38] [selection: [assignment: parts of TSF], the TSF]
[39] [selection: [assignment: parts of TSF data], TSF data]
[40] [selection: [assignment: parts of TSF], TSF]
[41] [assignment: physical tampering scenarios]
[42] [assignment: list of TSF devices/elements]
[43] [assignment: list of TSF devices/elements for which active detection is required]
[44] [assignment: a designated user or role]

Refinement: The management detection functionality scope (the devices /elements) is not necessary since if any device or element is selected within FPT_PHP.2.3 then its detection will be active for the entire life of the TOE. And also the notified entity will be the Embedded OS for the entire life of the TOE and there will be no other notified entity.

6.1.4 PROTECTION FROM ENVIRONMENTAL STRESS

FPT_FLS.1 protects the TOE from environmental stress.

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 operating conditions which may not be tolerated* [45].

6.1.5 ABUSE OF THE TEST FUNCTIONS

FMT_MOF.1/Test requires the TOE to offer the manufacturer to test functions disabling mechanism.

FMT_MOF.1/Test Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles fulfilled by FMT_SMR.1

FMT_SMF.1 Specification of Management Functions fulfilled by FMT_SMF.1

FMT_MOF.1.1 The TSF shall restrict the ability to *disable*[46] the functions *the manufacturing test functions*[47] to *the Manufacturer*[48].

Refinement: Once test functions are disabled, the TSF should irreversibly and permanently disable the test functions, so that their abuse during the usage of the TOE by embedded OS is not possible.

6.1.6 IC UNIQUE IDENTIFICATION DATA

The policy of IC Identification Data requires the TOE to be uniquely identified, FMT_MTD.1/ID requires the TOE to have Identification Data management function which enables to the TOE be uniquely identified. And also FMT_MTD.1/ID enables the protection of Identification Number from unauthorized manipulation.

FMT_MTD.1/ID Management of TSF data - Identification

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles fulfilled by FMT_SMR.1

[45] [assignment: list of types of failures in the TSF]
[46] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[47] [assignment: list of functions]
[48] [assignment: the authorised identified roles]

FMT_SMF.1 Specification of Management Functions fulfilled by FMT_SMF.1

FMT_MTD.1.1 The TSF shall restrict the ability to *write once*[49] the *IC Identification Data*[50] to the *Manufacturer*[51].

6.1.7 RANDOM NUMBER GENERATION

FCS_RND.1 fulfills the Random Number Generation Policy.

FCS_RND.1 Quality metric for random numbers

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: a defined quality metric][52].

6.1.8 ADDITIONAL SFRS FOR THE FLASH LOADER CONFIGURATION

FIA_AFL.1 protects the flash loader authentication mechanism from attacks.

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication fulfilled by FIA_UAU.1

FIA_AFL.1.1 The TSF shall detect when [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]][53] unsuccessful authentication attempts occur related to *Flash Loader Authentication*[54].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [selection: met, surpassed][55], the TSF shall [assignment: list of actions][56].

FIA_UAU.1 disables all functions except the ones that can be performed only once. Since IC Identification Data writing and Test Functions disabling can be performed only once, their operation can be performed without any user authentication.

[49] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[50] [assignment: list of TSF data]
[51] [assignment: the authorised identified roles]
[52] [assignment: a defined quality metric]
[53] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[54] [assignment: list of authentication events]
[55] [selection: met, surpassed]
[56] [assignment: list of actions]

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification fulfilled by FIA_UID.1

FIA_UAU.1.1 The TSF shall allow *IC Identification data write and Test Functions disable*[57] 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.5 requires the TOE to have user authentication mechanism to use the flash loader mechanism.

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1 The TSF shall provide *Flash Loader Authentication Mechanism*[58] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the *Flash Loader Authentication Mechanism authenticates the Card Issuer or the Manufacturer*[59].

FIA_UID.1 performs similar to FIA_UAU.1.

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow *IC Identification data write and Test Functions disable* [60] 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.

FMT_MTD.1/ES Management of TSF data – Embedded OS

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles fulfilled by FMT_SMR.1

[57] [assignment: list of TSF mediated actions]
[58] [assignment: list of multiple authentication mechanisms]
[59] assignment: rules describing how the multiple authentication mechanisms provide authentication]
[60] [assignment: list of TSF-mediated actions]

FMT_SMF.1 Specification of Management Functions fulfilled by FMT_SMF.1

FMT_MTD.1.1 The TSF shall restrict the ability to *write*[61] the *Embedded OS*[62] to the *Manufacturer and the Card Issuer*[63].

FMT_MOF.1/FL Management of security functions behavior – Flash Loader

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles fulfilled by FMT_SMR.1

FMT_SMF.1 Specification of Management Functions fulfilled by FMT_SMF.1

FMT_MOF.1.1 The TSF shall restrict the ability to *disable*[64] the functions *Embedded OS Loading*[65] to *Manufacturer and the Card Issuer*[66].

The first one of above two SFRs enable the writing of ES Data and the second one enable the disabling of the ES writing function since it might be abused by the attackers.

6.1.9 SECURITY MANAGEMENT FUNCTIONS AND ROLES

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:

- *Disabling the manufacturing test functions*
- *Write once the IC Identification data*
- *Embedded OS Loading*
- *Embedded OS Loading Locking*
- [assignment: list of management functions to be provided by the TSF][67].

Application Note:

Embedded OS Loading, Embedded OS Loading Locking is valid for Flash Loader Configuration For the ROM based configuration they do not exist.

FMT_SMR.1 Security roles

[61] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[62] [assignment: list of TSF data]
[63] [assignment: the authorised identified roles]
[64] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[65] [assignment: list of functions]
[66] [assignment: the authorised identified roles]
[67] [assignment: list of management functions to be provided by the TSF]

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification not fulfilled but justified

FMT_SMR.1.1 The TSF shall maintain the roles

- *Manufacturer*
- *Card Issuer*
- *Embedded OS*
- [assignment: the authorised identified roles][68].

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

Application Note:

Card Issuer role exists only for Flash Loader configuration if the embedded OS is loaded by the Card Issuer, otherwise this role does not exist.

The above two SFRs define the management functions and management roles needed for the TOE.

6.2 ASSURANCE REQUIREMENTS

Security Assurance Requirements for the TOE for the evaluation of the TOE are those taken from the Evaluation Assurance Level 5 (EAL5) and augmented by taking the following components:

- ALC_DVS.2 Sufficiency of security measures
- AVA_VAN.5. Advanced methodical vulnerability analysis

6.3 SECURITY REQUIREMENTS RATIONALE

6.3.1 RATIONALE FOR THE SECURITY FUNCTIONAL REQUIREMENTS

The following table provides an overview for security functional requirements coverage.

No:	Security Objective	Security Functional Requirements
1.	OT.Data_Access_Control	FDP_ACC.2, FDP_ACF.1, FCS_CKM.1/DP, FCS_CKM.4, FCS_COP.1/DP, FPT_PHP.3, FPT_PHP.2
2.	OT.Data_Integrity	FDP_SDI.2, FPT_TST.1, FPT_PHP.3, FPT_PHP.2 **Plus the SFRs for the OT.Data_Access_Control** FDP_ACC.2, FDP_ACF.1, FCS_CKM.1/DP, FCS_CKM.4, FCS_COP.1/DP
3.	OT.Internal_TOE_Transfer_Confidentiality_Protection	FDP_ITT.1

[68] [assignment: the authorised identified roles]

		FCS_CKM.1/TP, FCS_CKM.4, FCS_COP.1/TP FPT_PHP.3, FPT_PHP.2
4.	OT.Internal_TOE_Transfer_Integrity_Protection	FDP_ITT.3 FPT_PHP.3, FPT_PHP.2 **Plus the SFRs for the OT.Internal_TOE_Transfer_Confidentiality_Protection** FDP_ITT.1 FCS_CKM.1/TP, FCS_CKM.4, FCS_COP.1/TP
5.	OT.CPU_Operation_Protection	FPT_TST.1
6.	OT.Random_Number_Generation_Protection	FPT_TST.1
7.	OT.Side_Channel_Protection	FPT_SCP.1
8.	OT.Environmental_Stress_Protection	FPT_FLS.1
9.	OT.Test_Functions_Disable_Mechanism,	FPT_MOF.1/Test, FMT_SMF.1, FMT_SMR.1
10.	OT.Unique_ID_Storage	FMT_MTD.1/ID, FMT_SMF.1, FMT_SMR.1
11.	OT.Random_Number_Generation	FCS_RND.1

Table 3: Security Functional Requirements Rationale

ADDITIONAL RATIONALE FOR THE FLASH LOADER CONFIGURATION

12.	OT.Flash_Loader_Functionality	FPT_MTD.1/ES, FMT_SMF.1, FMT_SMR.1
13.	OT.Flash_Loader_Authorization,	FIA_AFL.1, FIA_UAU.1, FIA_UAU.5, FIA_UID.1
14.	OT.Flash_Loader_Disable_Mechanism	FPT_MOF.1/FL, FMT_SMF.1, FMT_SMR.1

Table 4: Security Functional Requirements Rationale

OT.Data_Access_Control

FDP_ACC.2 and FDP_ACF.1 provides access to the embedded OS while prevents the attacker. This is performed by storing the data encrypted and decrypting it with support of FCS_CKM.1/DP, FCS_CKM.4 and FCS_COP.1/DP. FPT_PHP.3 and FPT_PHP.2 provides additional protection against physical attacks.

OT.Data_Integrity

FDP_SDI.2 and FPT_TST.1 monitors the integrity of the data and performs the actions determined by the ST writer. FPT_PHP.3 and FPT_PHP.2 provides additional protection against physical attacks. The SFRs covering OT.Data_Access_Control is also valid for this objective, making reasonable changes to the encrypted data is not possible.

OT.Internal_TOE_Transfer_Confidentiality_Protection

FDP_ITT.1 provides protection while the data is in transmit between internal parts of the TOE. Protection is performed by encrypting the internal data traffic; encryption and decryption operations are performed by FCS_CKM.1/TP, FCS_CKM.4, FCS_COP.1/TP requirements. FPT_PHP.3 and FPT_PHP.2 provides additional protection against physical attacks.

OT.Internal_TOE_Transfer_Integrity_Protection

FDP_ITT.3 monitors the integrity of the data transferred and performs the actions determined by the ST writer. FPT_PHP.3 and FPT_PHP.2 provides additional protection against physical attacks. The SFRs covering OT.Internal_TOE_Transfer_Confidentiality_Protection is also valid for this objective, making reasonable changes to the encrypted data is not possible.

OT.CPU_Operation_Protection

FPT_TST.1 includes the test of correct operation of CPU, so covers the OT.CPU_Operation_Protection.

OT.Random_Number_Generation_Protection

FPT_TST.1 includes the test of correct operation of the Random Number Generation Functionality. So it covers the OT.Random_Number_Generation_Protection.

OT.Side_Channel_Protection

FPT_SCP.1 protects the TOE from side channel attacks.

OT.Environmental_Stress_Protection

FPT_FLS.1 require that if the TOE encounters environmental stress that it may not handle, it will preserve the secure state.

OT.Unique_ID_Storage

FMT_MTD.1 provides the functionality to the manufacturer to write the IC Identification data. So OT.Unique_ID_Storage is covered.

OT.Random_Number_Generation

FCS_RND.1 provides the random number generation functionality that OT.Random_Number_Generation states.

OT.Test_Functions_Disable_Mechanism

FMT_MOF.1/FL enables the functionality to irreversebly disable the test functionality. FMT_SMF.1 and FMT_SMR.1 defines this functionality and related roles.

ADDITIONAL RATIONALE FOR THE FLASH LOADER CONFIGURATION

OT.Flash_Loader_Functionality

FPT_MTD.1/ES enables the embedded operating system loading function. FMT_SMF.1 and FMT_SMR.1 defines this functionality and related roles.

OT.Flash_Loader_Authorization

FIA_UAU.5 requires a Flash Loader Authentication mechanism exist and FIA_UAU.1 and FIA_UID.1 require that Flash Loader operation can be performed before identification and authentication of the user. Finally FIA_AFL.1 protect the Flash Loader authentication mechanisms from false authentication attempts.

OT.Flash_Loader_Disable_Mechanism

FMT_MOF.1/FL enables the functionality to irreversebly disable the flash loader mechanism. FMT_SMF.1 and FMT_SMR.1 defines this functionality and related roles.

6.3.2 DEPENDENCIES FOR THE SECURITY FUNCTIONAL REQUIREMENTS

SFR	Dependencies	Support of the Dependencies
FDP_ACC.2	FDP_ACF.1	fulfilled by FDP_ACF.1/PAP
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	fulfilled by FDP_ACC.2/PAP justified (see justification 1)
FDP_ITT.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	fulfilled by FDP_ACC.1/PAP
FCS_CKM.1/PAP	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	fulfilled by FCS_COP.1 fulfilled by FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	fulfilled by FCS_CKM.1
FCS_COP.1/PAP	[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	fulfilled by FCS_CKM.1 fulfilled by FCS_CKM.4
FDP_SDI.2	None	---
FDP_ITT.3	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FDP_ITT.1 Basic internal transfer protection	fulfilled by FDP_ACC.1/PAP fulfilled by FDP_ITT.1
FPT_TST.1	None	---
FPT_PHP.3	None	---
FPT_PHP.2	FMT_MOF.1 Management of security functions behaviour	not fulfilled but justified
FPT_FLS.1	None	---
FMT_MOF.1/Test	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	fulfilled by FMT_SMR.1 fulfilled by FMT_SMF.1

FMT_MTD.1/Identification	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	fulfilled by FMT_SMR.1 fulfilled by FMT_SMF.1
FCS_RND.1	None	---
FPT_SCP.1	None	----
FMT_SMF.1	None	---
FMT_SMR.1	FIA_UID.1	fulfilled by FIA_UID.1

Table 5: Dependencies for the SFRs

Justification 1: The FPT_ACF.1 applies to the all data. So security attribute management is not necessary.

Justification 2: The management detection functionality scope (the devices /elements) is not necessary since if any device or element is selected within FPT_PHP.2.3 then its detection will be active for the entire life of the TOE. And also the notified entity will be the Embedded OS for the entire life of the TOE and there will be no other notified entity.

ADDITIONAL FOR FLASH LOADER

SFR	Dependencies	Support of the Dependencies
FIA_AFL.1	FIA_UAU.1 Timing of Authentication	fulfilled by FIA_UAU.1
FIA_UAU.1	FIA_UID.1 Timing of identification	fulfilled by FIA_UID.1
FIA_UAU.5	None	----
FIA_UID.1	None	----
FMT_MTD.1/ES	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	fulfilled by FMT_SMR.1 fulfilled by FMT_SMF.1
FMT_MOF.1/FL	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	fulfilled by FMT_SMR.1 fulfilled by FMT_SMF.1

Table 6: Dependencies for the Additional SFRs of the Flash Loader

6.3.3 RATIONALE AND DEPENDENCIES FOR THE SARS

The TOE within this PP aims to withstand attackers with attack potential of HIGH; so besides the functional requirements, the assurance requirements: architectural soundness and well defined and tested internals are important. EAL5 package requires semiformal design descriptions, a more structured (and hence analyzable) architecture, internal focused testing which EAL4 does not require. So assurance requirements for the TOE and EAL5 match. Since reverse engineering is also an important treat for the TOE and the required assurance from Development Environment is high; ALC_DVS.2 is added. Finally to meet withstand attackers with HIGH attack potential AVA_VAN.5 added.

EAL5 to EAL4 Differences

EAL4	**EAL5**
ADV_FSP.4	ADV_FSP.5
---	ADV_INT.2
ADV_TDS.3	ADV_TDS.4
ALC_CMS.4	ALC_CMS.5
ALC_TAT.1	ALC_TAT.2
ATE_DPT.1	ATE_DPT.3
AVA_VAN.3	AVA_VAN.4

Table 7: Differences between EAL4 and EAL5

The dependencies for augmented SARs:

SAR	**Dependencies**	**Support of the Dependencies**
ALC_DVS.2	None	---
AVA_VAN.5	ADV_ARC.1 Security architecture description ADV_FSP.4 Complete functional specification ADV_TDS.3 Basic modular design ADV_IMP.1 Implementation representation of the TSF AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures ATE_DPT.1 Testing: basic design	Included in EAL5

Table 8: The Dependencies for augmented SARs

6.3.4 SECURITY REQUIREMENTS – MUTUAL SUPPORT AND INTERNAL CONSISTENCY

Current PP aims to withstand against attacker of HIGH attack potential. Both the SFRs and the SARs are selected to reach this goal. The rationale of both requirement types (functional requirements and assurance requirements are given and dependency analysis of them are made; no inconsistency exists. The SFRs and SARs internally support each other.

The support for the SFRs and SFRs of each other is such that SARs are sufficient to give enough assurance for the required functionality.

7. GLOSSARY AND ACRONYMS

CPU: Central Processing Unit

ES: Embedded Operating System

I/O: Input Output

OS: Operating System

ROM: Read only memory

RAM: Random Access Memory

SAR: Security Assurance Requirement

SFR: Security Functional Requirement

TOE: Target of Evaluation

TSF: TOE Security Functionality

8. REFERENCES

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 1, September 2012

[4] Common Methodology for Information Technology Security Evaluation (CEM), Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012