BAROC Smart Card Protection Profile

Version: 1.2

Date: 2005-11-11

Authors: BAROC/FISC Smart Card Group

Table of contents

List of tables

List of figures

1 PP introduction

1.1 PP identification

Title:	Financial Smart Card Application Protection Profile
TOE class:	Financial Smart Card for the Taiwanese Market
Document name:	PP_FISC_V1.2
Version:	1.2
Document Date:	2005-11-11
Author:	BAROC/FISC Smart Card Group
CC Version	2.1
	All final interpretations until September, 20th 2005 have been considered
EAL:	4+ augmented by AVA_VLA.4 and ADV_IMP.2
SOF-claim:	high
Certification ID:	BSI-PP-0021
Evaluation Body:	TÜViT GmbH, Germany
Certification Body:	BSI, Germany
Keywords:	Smart card, TAC, BAROC, financial transaction, FISC, Taiwan Banking System, Common Criteria, Protection Profile

1.2 PP overview

Because of serious circumstances of counterfeiting and skimming, and because of the functional limitations of magnetic stripe cards, the Bankers Association of the Republic of China (BAROC) initiated the Chip Migration Task Force Team in Feb. 2001, to evaluate the feasibility of Chip Migration Project and to develop related specifications.

BAROC developed this Protection Profile to serve as a baseline for the security of smartcards developed by different vendors. These smartcards will be used for the financial transactions within the FISC inter-bank system.

This PP focuses on a Financial Smart Card which consists of embedded software and a secure IC Controller. The TOE is used as a security token for inter-bank financial transactions, such as cash withdrawal, fund transfer, tax payment and online sale.

The main objectives of this Protection Profile are:

- To describe the security environment of the TOE including assets to be protected and threats to be countered by the TOE and its environment.
- To describe the security objectives of the TOE and its supporting environment.
- To specify the security requirements, which include the TOE security functional requirements and the assurance requirements

1.3 CC conformance claims

This PP is claimed to be [CC] part 2 extended (FPT_EMAN.1) and [CC] part 3 conformant. This PP does not claim conformance to any other PP. The CC version used is: ISO/IEC 15408: Common Criteria, Version 2.1 All final interpretations until September, 20th 2005 have been considered.

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

The assurance level is EAL4 augmented by AVA_VLA.4 (highly resistant) and ADV_IMP.2 (Implementation of the TSF).

1.4 Acknowledgement

The authors would like to highlight the significant impact of [SSCD] to the development of this Protection Profile. Due to the special requirements for the Taiwanese Financial Market it has unfortunately not been possible to directly use [SSCD]. Nevertheless many of the requirements for this PP and especially the extension of CC part II with FPT_EMAN.1 have been taken from or inspired by the requirements in [SSCD].

2 TOE description

2.1 Overview

The TOE is a smart card which consists of embedded software and a secure IC Controller. The main purpose of the TOE is to act as a token in the FISC Inter-bank System (see Figure 2.1) where a cardholder can do financial transactions such as cash withdrawal, fund transfer, tax payment and purchase with it. The FISC Inter-bank System is a general-purpose platform for switching financial transactions between banks. The FISC Inter-bank System includes Issuer Bank, FISC, Acquire Bank and its Card Accepted Devices (CAD). The Issuer Bank is in charge of issuing cards to customers and authorizing online transactions from customers. FISC is in charge of switching, clearing and settlement of financial transactions. The Acquire Bank is in charge of Card Accepted Devices or so-called application channels and acquiring transactions from aforementioned application channels. The Issuer Bank and Acquire Bank shall be recognized by FISC.

Figure 1: Inter-Bank-System

Take fund transfer as an example; the transaction flow is as following:

1. A cardholder inserts its smartcard into the CAD and enters its PIN
2. The cardholder selects the “fund transfer” function.
3. The cardholder confirms the transaction. The CAD prepares transaction data characteristic for the type of transaction and sends it to the TOE via APDU command (following [ISO7816] part 4, augmented with TAC generation).
4. The TOE generates a serial number and a TAC in response to the CAD request.
5. The serial number and TAC are then transmitted to Issuer Bank via the FISC inter-bank system for transaction approval.
6. If the transaction is approved by Issuer Bank, the transaction amount is transferred.

2.2 TOE Definition

The TOE is composed of a Smart Card IC and embedded software. Within the Taiwanese banking system aforementioned, the TOE is used to secure financial transactions.

Therefore, the TOE is able to generate a transaction authentication code (TAC) for a transaction record (also called DTBT = Data to be TAC'd) which is representing a kind of digital signature to secure the authenticity and integrity of the transaction.

Within this system, the major scope of the TOE is to protect the key which is used to generate a TAC.

2.3 TOE Boundaries

2.3.1 Physical Boundary

The TOE consists of a SmartCard with a physical interface compliant to ISO 7816 part 2 with its dedicated software as well as the SmartCard embedded software and the related guidance documentation.

2.3.2 Logical Boundary

The TOE logical interface is represented by a set of APDU commands which is compliant to ISO 7816 part 4 (augmented with additional commands).

At its logical boundary, the TOE provides functions to generate a TAC for DTBT which can be sent to the TOE. The TOE provides no possibility to read out any cryptographic key but only to update the key which is used for TAC generation.

The TOE is acting as a kind of signature token which produces a TAC for every DTBT which is sent to the TOE. Before TAC generation, the user has to enter a PIN to confirm the TAC generation. However, disclosure of a confirmation PIN during entry by the CAD is not considered as a threat, and therefore, no trusted channels have to be provided by the TOE.

2.4 TOE Life Cycle

The TOE life cycle (LC) is shown in the following figure.

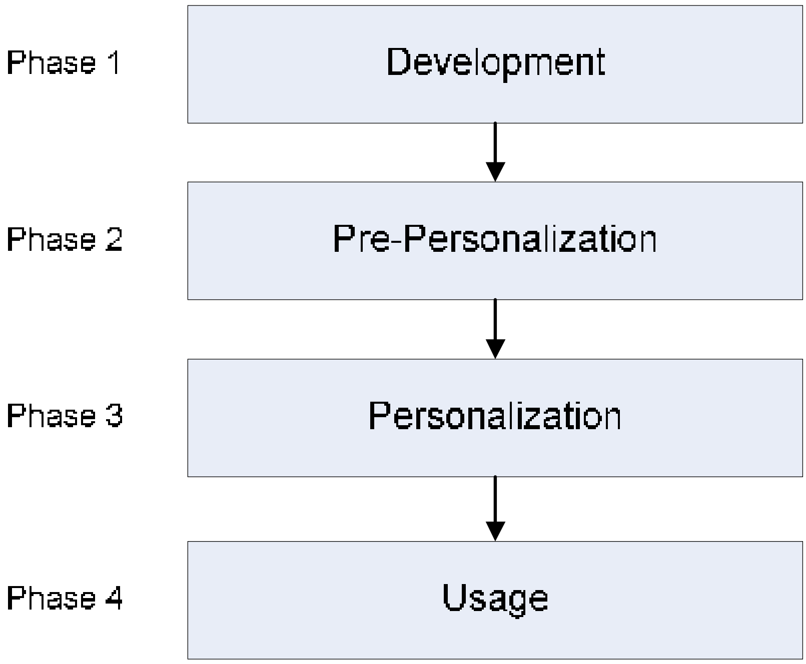

Figure 2: Financial Smart Card Application life cycle

The stages shown are listed below:

Phase 1: This phase covers the development and production process of the hardware and software the TOE is consisting of.

Phase 2: During the Pre-personalization process, the TOE is initialized. This is typically done at the site of card manufacturer. The delivery is done in a secure manner after this phase.

Phase 3: This phase includes provisioning all user data into the TOE which is necessary for the usage. This process is typically done at the site of issuing bank.

Phase 4: The cardholder can use the TOE to secure financial transactions via the FISC Inter-bank System.

2.5 Roles

The TOE maintains the following roles:

- Administrator — An administrator is the only role which is allowed to use the key update functionality of the TOE provided during the phases 3 and 4.
- Cardholder — A cardholder is a person who handles the TOE in usage phase. The person who holds the TOE is allowed to use it to generate a TAC in phase 4 (see TOE Life Cycle).

2.6 Description of TOE security functionality

The TOE security functionality consists of TAC generation, secure key update, and protection of TSF and user data.

2.6.1 TAC generation

The TOE calculates a TAC (Transaction Authentication Code) on transaction data. The TAC ensures authenticity and integrity of the transaction data. In addition to the TAC, the TOE also generates a transaction S/N (serial number) which participates in the calculation of the TAC. In order to generate a TAC, the user has to enter a PIN for confirmation.

2.6.2 Secure key update

The TOE is providing a secure means to update cryptographic keys (especially the key which is used for TAC generation) that will be stored in the TOE.

2.6.3 Protection of TSF and user data

The TOE protects its TSF and user data from unauthorized modification and disclosure.

3 TOE security environment

3.1 Assets

Assets are security relevant elements of the TOE. Generally speaking, the following groups of assets are available:

- Embedded software including specifications, implementation and related documentation
- Application data of the TOE (e.g. IC and software specific data, Initialisation data, Personalisation data)

3.1.1 TAC Key

The TAC (Transaction Authentication Code) Key is a cryptographic key and is used by the “TAC Generation” within the TOE. The TAC key is stored in the EEPROM of the IC Controller during Phase 3. The TOE has to ensure the integrity and confidentiality of the TAC Key.

3.1.2 Perso and Pre-perso Data

This data consists of user data and cryptographic keys.

3.1.3 PIN

The PIN (Personal Identification Number) of the TOE is used to authenticate the user of the TOE. The PIN length shall be at least 6 digits and can be up to 12 digits. The PIN is initially generated and stored in the EEPROM of IC controller by the administrator during Phase 3, and can be changed by Cardholder and Administrator during Phase 4. The TOE has to ensure the integrity and confidentiality of the PIN when stored on the card.

3.1.4 Retry Counter

There is one retry counter stored in the EEPROM of IC Controller during Phase 2-4. It is for accumulating consecutive failure attempts of Terminal Authentication and User Authentication. The status is blocked as the Retry Counter reaches the Retry Limit. The TOE has to ensure the integrity and confidentiality of the Retry Counter (Phase 2-4).

3.1.5 Retry Limit

An upper bound of the Retry Counter stored in the EEPROM of IC Controller by Issuer Bank during Phase 3 to prohibit further attempts of authentication when the Retry Counter reaches its associated Retry Limit. The TOE has to ensure the integrity of the retry limit (Phase 2-3).

3.1.6 Serial Number for transactions

A number which is incremented automatically by TOE after each transaction. It participates in TAC generation to ensure that the TAC calculation is not only based on DTBT but also based on the serial number.

3.1.7 DTBT (Data To Be TAC'd)

This is the data which is received by the TOE to generate a TAC over. In the case of this TOE the DTBT is a transaction record which is used to secure a financial transaction.

3.2 Assumptions (about the environment)

Assumption name	Description
A.PERSO	The Personalization and Pre-Personalization process is assumed to take place in an environment providing adequate physical security and performed by trustworthy personnel. Any data which is handled during these processes must be kept confidential. During key update, a secure CAD which is able to provide authentication and encryption has to be used.
A.KEY	All cryptographic keys which are created in the environment to be used within the TOE have to be created and handled in a secure manner and must have sufficient quality.
A.DEVELOPMENT	TOE development and test information during phases 1 and 2 is protected in a secure environment for its integrity and confidentiality. In case of delivery between different actors like IC manufacturer and embedded software developer, this information is also protected in the same manner as aforementioned.

Table 1: Assumptions

3.3 Threats

The threats in this chapter have been developed based on the following definition of an attacker:

An attacker is a person who is trying to access sensitive information. His motivation is to get able to copy or clone the TOE to compromise the whole financial system which is secured by the TOE. However misuse of one single TOE in the way of generating a TAC without the authorization of the owner of the card is not considered as an attack. To perform his attack, the attacker has access to nearly unlimited resources in terms of money and time. Therefore the attacker has a high attack potential in terms of CC.

Threat name	Description
T.HACK_PHYS *Physical attacks through the TOE interfaces*	An attacker may obtain knowledge of cryptographic keys via physical attacks such as probing.
T.LEAKAGE *Leakage of information from the TOE*	An attacker may obtain TSF-data which is leaked from the TOE during normal usage. Leakage of information may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by

Threat name	Description
	changes in processing time requirements.
T.KEY_COMPROMISE *Copying, releasing or unauthorized modification of the cryptographic keys*	An attacker may try to compromise the secret cryptographic key of the TOE. He may try to copy secret keys from the TOE using the user visible interfaces of the TOE. He may also try to use a brute force attack against the authentication mechanism of the administrator to overwrite or delete the key. An attacker may try to perform this attack during the usage phase of the TOE or during the key update process.
T.KEY_DERIVE *Derive the TAC key*	An attacker derives the TAC key from public known data, such as a TAC created by means of the TAC key or any other data communicated outside the TOE, which is a threat against the secrecy of the TAC key.
T.INTEGRITY *Integrity of security relevant data*	An attacker may change security relevant data in the storage of the TOE. Security relevant data includes cryptographic keys, TAC and DTBT.

332 **Table 2: Threats**

333 3.4 Organisational security policies

OSP Name	Description
OSP.TAC	The TOE has to provide a function to generate a TAC over a DTBT. The TOE has to use a cryptographic operation to generate the TAC with the TAC key. The TAC is comparable to a digital signature while as the DTBT to the data to be signed. The TAC generation has to include an automatically incremented unique serial number. The serial number participates in the TAC generation process to achieve that TAC calculation is not only based on DTBT but also the serial number.
OSP.PIN	In order to use the “TAC Generation” function of the TOE, the user of the TOE has to enter a PIN beforehand. This PIN is primarily thought of as a confirmation from the user. To perform more than one transaction the user has to enter the PIN only one time. The TOE shall not provide any function to read out the PIN.
OSP.KEY_UPDATE	The TOE has to provide a secure communication channel and authentication to update cryptographic keys in a secure manner.

334 **Table 3: Organisational Security Policies**

4 Security Objectives

4.1 Security objectives for the TOE

Objective Name	Description
SO.EMAN_DESIGN *Provide physical emanations security*	The TOE has to be designed and built in such a way as to control the production of intelligible emanations within specified limits.
SO.SELF_TEST *Self Testing*	The TOE shall provide self-testing functionality for all TOE security functions which can detect flaws during pre-personalisation, personalisation and operational usage phases.
SO.KEY_SECRECY *Secrecy of the cryptographic keys*	The secrecy of *cryptographic keys* (e.g. the TAC key that is used for TAC generation) is reasonably assured against attacks with a high attack potential.
SO.TAMPER_ID *Tamper detection*	The TOE provides system features that detect physical tampering of a system component.
SO.TAMPER_RESISTANCE *Tamper resistance*	The TOE prevents or resists physical tampering with specified system devices and components.
SO.KEY_UPDATE *Secure updates of the cryptographic keys*	The TOE has to provide a secure mechanism to update *cryptographic keys*. This includes mechanisms to ensure the confidentiality and integrity of *cryptographic keys* transferred to the TOE as well as the authentication of the terminal which is sending the keys. The TOE shall provide safe destruction techniques for the cryptographic keys in case of key updates.
SO.TAC_CONFIRM *TAC generation function after confirmation only*	The TOE provides the TAC generation function only after the user has entered his PIN for confirmation. For multiple TAC generations the user has to enter the PIN only one time. The TOE must not provide a function which would allow anybody to read out the PIN.
SO.TAC_SECURE *Cryptographic security of the TAC*	The TOE generates a TAC that cannot be forged without access to the TAC key through robust encryption techniques. The TAC key must not be reconstructible from publicly available data, such as a TAC or its DTBT. The TAC generation includes an automatically incremented unique serial number. The serial number participates in the TAC generation process to achieve that TAC calculation is not only based on DTBT but also based on this serial number.

SO.INTEGRITY *Integrity Protection*	The TOE protects data in its storage against any unauthorized modification.

Table 4: Security Objectives for the TOE

4.2 Security objectives for the environment

Objective name	Description
SOE.PERSO	The Personalization and Pre-Personalization process must take place in an environment providing adequate physical security and performed by trustworthy personnel. Any data which is handled during these processes must be kept confidential. During key update, a secure CAD which is able to provide authentication and encryption has to be used.
SOE.KEY	All cryptographic keys which are created in the environment to be used within the TOE have to be created and handled in a secure manner and have to have sufficient quality.
SOE.DEVELOPMENT	TOE development and test information during phases 1 and 2 is protected in a secure environment for its integrity and confidentiality. In case of delivery between different actors like IC manufacturer and embedded software manufacturer, this information is also protected in the same manner as aforementioned.

Table 5: Security Objectives for the environment

5 IT Security Requirements

This chapter gives the security functional requirements and the security assurance requirements for the TOE and the environment.

Security functional requirements components given in section 5.1 “TOE security functional requirements”, excepting FPT_EMAN.1 which is explicitly stated, are drawn from Common Criteria part 2 [CC]. Operations for assignment and selection have been made. Operations not performed in this PP are identified in order to enable instantiation of the PP to a Security Target (ST).

All operations which have been performed from the original text of part 2 of [CC] are written in *italics* for assignments and underlined for selections. Furthermore the [brackets] from part 2 of [CC] are kept in the text.

All operations which have to be completed by the ST author are marked with the words: "assignment" or "selection" respectively.

The TOE security assurance requirements statement given in section 5.2 “TOE Security Assurance Requirement” is drawn from the security assurance components from Common Criteria part 3 [CC].

Section 5.3 identifies the IT security requirements that are to be met by the IT environment of the TOE.

The non-IT environment is described in section 5.4

5.1 TOE Security Functional Requirements

The following table provides an overview about the used SFRs:

SFR	Description
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1	Cryptographic operation
FDP_ACC.1/KEY	Subset access control for cryptographic keys
FDP_ACC.1/TAC	Subset access control for the TAC generation
FDP_ACF.1/KEY	Security attribute based access control for cryptographic keys
FDP_ACF.1/TAC	Security attribute based access control for the TAC generation
FDP_ITC.1	Import of user data without security attributes
FDP_RIP.1	Subset residual information protection
FDP_SDI.2	Stored data integrity monitoring and action
FDP_UCT.1	Basic data exchange confidentiality
FDP_UIT.1	Data exchange integrity
FIA_AFL.1/PIN	Authentication failure handling regarding the PIN
FIA_AFL.1/KEY	Authentication failure handling regarding the Key
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UAU.5	Multiple authentication mechanisms
FIA_UID.1	Timing of identification
FMT_MSA.1/TAC	Management of security attributes for TAC
FMT_MSA.1/KEY	Management of security attributes for keys
FMT_MSA.2	Secure security attributes
FMT_MSA.3/TAC	Static attribute initialisation for TAC
FMT_MSA.3/KEY	Static attribute initialisation for keys
FMT_MTD.1	Management of TSF data
FMT_SMF.1/PIN	Specification of Management Functions for PIN
FMT_SMF.1/KEY	Specification of Management Functions for TAC
FMT_SMR.1	Security roles
FPT_AMT.1	Abstract machine testing
FPT_EMAN.1	TOE Emanation
FPT_FLS.1	Failure with preservation of secure state
FPT_PHP.1	Passive detection of physical attack
FPT_PHP.3	Resistance to physical attack
FPT_TST.1	TSF testing
FTP_ITC.1	Inter-TSF trusted channel

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic key destruction (FCS_CKM.4)

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

Application Note: It must be assured that cryptographic keys are destroyed securely e.g. by overwriting by new keys.

5.1.1.2 Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform [*TAC generation including a unique transaction serial number*] in accordance with a specified cryptographic algorithm [*assignment: cryptographic algorithm*] and cryptographic key sizes [*assignment: cryptographic key sizes*] that meet the following: [*listed in [FIPS_A]*].

Application Note: TAC shall include an automatically incremented unique serial number. The serial number participates in the TAC generation process to achieve that TAC calculation is not only based on DTBT but also based on the serial number.

5.1.2 User data protection (FDP)

5.1.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1/KEY The TSF shall enforce the [*Key Import/export SFP*] on [*subjects: user, objects: cryptographic keys and operation: import and export of keys*].

FDP_ACC.1.1/TAC The TSF shall enforce the [*TAC Generation SFP*] on [*subjects: user, objects: DTBT and operation: generate a TAC*].

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1/KEY The TSF shall enforce the [*Key Import/export SFP*] to objects based on the following: [*subject attribute: Administrator {yes/no} and object attribute: cryptographic key {yes/no}*].

FDP_ACF.1.2/KEY The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*users with subject attribute administrator set to {yes} are allowed to update objects with attribute cryptographic key set to {yes}*].

FDP_ACF.1.3/KEY The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*no other rule*].

FDP_ACF.1.4/KEY The TSF shall explicitly deny access of subjects to objects based on the [*rules:*

Nobody is allowed to read out objects with attribute secret key set to {yes}].

FDP_ACF.1.1/TAC The TSF shall enforce the [*TAC Generation SFP*] to objects based on the following: [*subject attribute: Cardholder {yes/no}, object attribute PIN {yes/no}*].

FDP_ACF.1.2/TAC The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*users* *with subject attribute Cardholder set to {yes} are allowed to generate a TAC for DTBT sent to the TOE*].

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

FDP_ACF.1.4/TAC The TSF shall explicitly deny access of subjects to objects based on the [*nobody is allowed to read out an object with attribute PIN set {yes}*].

5.1.2.3 Import of user data without security attributes (FDP_ITC.1)

FDP_ITC.1.1 The TSF shall enforce the [*Key Import/export SFP*] when importing user data, controlled under the SFP, from outside of the TSC.

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

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [*The key must only be accepted when sent by an authorized administrator via the trusted channel*]

5.1.2.4 Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*selection: allocation of the resource to, deallocation of the resource from*] the following objects: [*cryptographic keys, PIN, [assignment: none or a list of objects]*].

5.1.2.5 Stored data integrity monitoring and action (FDP_SDI.2)

FDP_SDI.2.1 The TSF shall monitor user data stored within the TSC for [*assignment: integrity errors*] on all objects, based on the following attributes [*assignment: user data attributes*].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [

1. *Prohibit the use of the altered data*
2. *Inform the user about integrity errors*]

5.1.2.6 Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the [*Key Import/export SFP*] to be able to [receive] objects in a manner protected from unauthorised disclosure.

5.1.2.7 Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the [*Key Import/export SFP*] to be able to [receive] user data in a manner protected from [modification, insertion] errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether [modification, insertion] has occurred.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1/PIN The TSF shall detect when [an administrator configurable positive integer *within 1 to 15*] unsuccessful authentication attempts occur related to [*PIN based authentication of the Cardholder*].

FIA_AFL.1.2/PIN When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*block the PIN based authentication of the Cardholder*].

Application Note: Even though the PIN entry of the user is more seen as a confirmation mechanism than as to be an authentication mechanism, this mechanism is modelled using SFRs from class FIA.

FIA_AFL.1.1/KEY The TSF shall detect when [an administrator configurable positive integer *within 1 to 15*] unsuccessful authentication attempts occur related to [*Key based authentication of the Administrator*].

FIA_AFL.1.2/KEY When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*block the Key based authentication of the Administrator*].

Application Note: For the first assignment in FIA_AFL.1.1/PIN and FIA_AFL.1.1/KEY it would also be acceptable if the number of allowed unsuccessful authentication attempts is fixed and not configurable by the admin.

5.1.3.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [*PIN, Cardholder {yes/no}, Administrator {yes/no}, number of unsuccessful authentication attempts]*

5.1.3.3 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [*assignment: list of TSF mediated actions*] 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.

Application Note: The ST author must not specify one of the following TSF mediated actions in the assignment of FIA_UAU.1.1:

1. TAC generation
2. Key update
3. Management functions provided by the TOE

5.1.3.4 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide [*PIN based and Key based authentication mechanisms*] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [*PIN based authentication is used for authenticating a Cardholder and Key based authentication is used for authenticating an Administrator*].

5.1.3.5 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [*assignment: list of TSF-mediated actions*] 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.

Application Note: The ST author must not specify one of the following TSF mediated actions in the assignment of FIA_UID.1.1:

1. TAC generation
2. Key update
3. Management functions provided by the TOE

5.1.4 Security management (FMT)

5.1.4.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1/TAC The TSF shall enforce the [*TAC generation SFP*] to restrict the ability to [modify] the security attributes [*Cardholder {yes/no}*] to [*Cardholder*]

FMT_MSA.1.1/KEY The TSF shall enforce the [*Key Import/export SFP*] to restrict the ability to [query, [*set*]] the security attributes [administrator {yes/no}, *cryptographic key {yes/no}*] to [*administrator*].

5.1.4.2 Secure security attributes (FMT_MSA.2)

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

5.1.4.3 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1/TAC The TSF shall enforce the [*TAC generation SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/TAC The TSF shall allow the [*no roles*] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3.1/KEY The TSF shall enforce the [*Key Import/export SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/KEY The TSF shall allow the [*no roles*] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [modify] the [*PIN*] to [*Cardholder or Administrator*].

5.1.4.5 Specification of Management Functions(FMT_SMF.1)

FMT_SMF.1.1/PIN The TSF shall be capable of performing the following security management functions: [*Modify the PIN, Set number of unsuccessful authentication attempts*].

FMT_SMF.1.1/KEY The TSF shall be capable of performing the following security management functions: [*query and set the security attributes of cryptographic key, start the self test of the TOE*].

5.1.4.6 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [*Administrator and Cardholder*].

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

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Abstract machine testing (FPT_AMT.1)

FPT_AMT.1.1 The TSF shall run a suite of tests [during initial start-up, periodically during normal operation, at the request of an authorised user, [*assignment: other conditions*]] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

5.1.5.2 TOE Emanation (FPT_EMAN.1)

FPT_EMAN.1.1 The TOE shall not emit *[assignment: types of emissions]* in excess of *[assignment: specified limits]* enabling access to secret data including cryptographic keys, especially the TAC key.

FPT_EMAN.1.2 The TSF shall ensure that nobody is able to use *[assignment: types of emissions]* to gain access to secret data including cryptographic keys, especially the TAC key.

Application Note: The TOE shall prevent attacks against cryptographic keys and other secret data where the attack is based on external observable physical phenomena of the TOE. Such attacks may be observable at the interfaces of the TOE or may origin from internal operation of the TOE or may origin by an attacker that varies the physical environment under which the TOE operates. The set of measurable physical phenomena is influenced by the technology employed to implement the TOE. Examples of measurable phenomena are variations in the power consumption, the timing of transitions of internal states, electromagnetic radiation due to internal operation, radio emission. Due to the heterogeneous nature of the technologies that may cause such emanations, evaluation against state-of-the-art attacks applicable to the technologies employed by the TOE is assumed. Examples of such attacks are, but are not limited to, evaluation of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc.

5.1.5.3 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [*assignment: list of types of failures in the TSF*].

5.1.5.4 Passive detection of physical attack (FPT_PHP.1)

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

5.1.5.5 Resistance to physical attack (FPT_PHP.3)

FPT_PHP.3.1 The TSF shall resist *[assignment:* p*hysical tampering scenarios]* to the *[assignment: list of TSF devices/elements]* by responding automatically such that the TSP is not violated.

5.1.5.6 TSF testing (FPT_TST.1)

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: conditions under which self test should occur]*] to demonstrate the correct operation of the TSF.

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

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Application Note: According to SO.SELF_TEST, TOE self-test should be provided for pre-personalisation, personalisation and operational usage phases.

5.1.6 Trusted path/channels (FTP)

5.1.6.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit *[selection: the TSF, the remote trusted IT product]* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*import of cryptographic key,* [*assignment: any other functions for which a trusted channel is required*]].

5.2 TOE Security Assurance Requirements

The evaluation assurance package is EAL 4 augmented by AVA_VLA.4 and ADV_IMP.2.

5.3 Security Requirements for the IT Environment

5.3.1 Cryptographic key generation

5.3.1.1 Cryptographic key generation (FCS_CKM.1/ENV)

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

5.3.1.2 Basic data exchange confidentiality (FDP_UCT.1/ENV)

FDP_UCT.1.1/ENV The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to be able to [*transmit*] objects in a manner protected from unauthorised disclosure.

5.3.1.3 Data exchange integrity (FDP_UIT.1/ENV)

FDP_UIT.1.1/ENV The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to be able to [*transmit*] user data in a manner protected from [*modification*, *insertion*] errors.

FDP_UIT.1.2/ENV The TSF shall be able to determine on receipt of user data, whether [*modification*, *insertion*] has occurred.

5.3.1.4 Inter-TSF trusted channel (FTP_ITC.1/ENV)

FTP_ITC.1.1/ENV The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/ENV The TSF shall permit *[selection: the TSF, the remote trusted IT product]* to initiate communication via the trusted channel.

FTP_ITC.1.3/ENV The TSF shall initiate communication via the trusted channel for [*export of cryptographic key,* [*assignment: any other functions for which a trusted channel is required*]].

Note that the dependencies of the security requirements in the environment have not been considered.

To identify the SFRs mentioned in this chapter as SFRs for the environment the identifiers from part II of [CC] have been modified with a suffix.

5.4 Security Requirements for the Non-IT Environment

5.4.1 R.Personalization

The Personalization and Pre-Personalization process must take place in an environment providing adequate physical security and performed by trustworthy personnel.

Any data which is handled during these processes have to be kept confidential.

5.4.2 R.Key_Protection

All cryptographic keys which are created in the environment to be used within the TOE have to be handled in a secure manner.

5.4.3 R.Development

TOE development and test information during phases 1 and 2 must be protected in a secure environment for its integrity and confidentiality. In case of delivery between different actors like IC manufacturer and embedded software manufacturer, this information must be also protected in the same manner as aforementioned.

6 Rationale

6.1 Security objectives rationale

Threats, Assumptions, OSP / Security Objectives	SO.EMAN_DESIGN	SO.SELF_TEST	SO.KEY_SECRECY	SO.TAMPER_ID	SO.TAMPER_RESISTANCE	SO.KEY_UPDATE	SO.TAC_CONFIRM	SO.TAC_SECURE	SO.INTEGRITY	SOE.PERSO	SOE. KEY	SOE.DEVELOPMENT
T.HACK_PHYS				X	X							
T.LEAKAGE	X											
T.KEY_COMPROMISE		X	X			X				X		
T.KEY_DERIVE		X						X				
T.INTEGRITY		X							X			
OSP.TAC		X						X				
OSP.PIN		X					X					
OSP.KEY_UPDATE		X				X						
A.PERSO										X		
A.KEY											X	
A.DEVELOPMENT												X

Table 6: Security Objectives Rationale

6.1.1 Coverage of the Security Objectives

SO.EMAN_DESIGN can be traced back to the threats **T.LEAKAGE** as the design which is described in **SO.EMAN_DESIGN** prevents any emanations which could be used to perform **T.LEAKAGE**.

SO.SELF_TEST can be traced back to many threats as it is supporting all security functions which are provided by the TOE because it ensures that these functions are working correctly.

SO.KEY_SECRECY can be traced back to the threats **T.KEY_COMPROMISE** as **SO.KEY_SECRECY** describes that the confidentiality of the cryptographic keys has to be ensured by the TOE.

SO.TAMPER_ID can be traced back to the threats **T.HACK_PHYS** as one have to identify an attack via physical means before one is able to handle this attack.

SO.TAMPER_RESISTANCE can be traced back to the threats **T.HACK_PHYS** as **SO_TAMPER_RESISTANCE** defines that the TOE has to prevent or resist physical hacking as described in **T.HACK_PHYS.**

SO.KEY_UPDATE can be traced back to the threats **T.KEY_COMPROMISE** as it ensures that the confidentiality of the cryptographic key is ensured when transmitted to the TOE and **OSP.KEY_UPDATE** as this objective describes the functionality as required by the OSP.

SO.TAC_CONFIRM can directly be traced back to the **OSP.PIN**.

SO.TAC_SECURE can be traced back to **OSP.TAC** as it describes the requirements from the OSP and to the threat **T.KEY_DERIVE** as the mechanism as described in **SO.TAC_SECURE** are used to block the possibility to gain knowledge of the secret keys with public knowledge.

SO.INTEGRITY can obviously be traced back to **T.INTEGRITY**.

6.1.2 Coverage of the assumptions

A.PERSO is obviously covered by **SOE.PERSO**.

A.KEY is obviously covered by **SOE.KEY**.

A.DEVELPOPMENT is obviously covered by **SOE.DEVELOPMENT**.

All the security objectives for the environment are stated in a way that it is obvious that they are suitable to fulfil the assumption.

6.1.3 Countering the threats

SO.SELF_TEST is a supportive security objective which is enlisted against many threats. It will therefore not be explicitly mentioned in the following paragraphs. It ensures that the security functions which are provided by the TOE are working correctly and is therefore a supportive objective for all threats which are actively blocked by functions of the TOE.

T.HACK_PHYS is covered by **SO.TAMPER_ID** which detects physical tampering and **SO.TAMPER_RESISTANT** which requires that the TOE has to be resistant against this kind of attacks.

T.LEAKAGE is obviously covered by **SO_EMAN_DESIGN**.

T.KEY_COMPROMISE is covered by **SO.KEY_SECRECY** which secures the cryptographic keys when stored in the TOE and **SO.KEY_UPDATE** which protects the key when transmitted to the TOE. Furthermore **SOE.PERSO** supports the blocking of this threat as it ensures that the confidentiality of the key is ensured during the perso- or update process.

T.KEY_DERIVE is directly covered by **SO.TAC_SECURE** as this objective defines that any algorithm which is used to calculate the TAC has to ensure that it is not feasible to derive the secret key from any publicly available data.

T.INTEGRITY is directly covered by **SO.INTEGRITY** as it is not feasible for an attacker to change any kind of security relevant data as long as the TOE protects its data against unauthorized modification.

6.1.4 Coverage of the Organisational Security Policies

OSP.TAC is obviously covered by **SO.TAC_SECURE**.

OSP.PIN is obviously covered by **SO.TAC_CONFIRM**.

OSP.KEY_UPDATE is obviously covered by **SO.KEY_UPDATE**.

All these security objectives are stated in a way that it is obvious that they are suitable to fulfil the OSP.

6.2 Security requirements rationale

6.2.1 Suitability of minimum strength of function (SoF) level

The TOE shall be highly resistant against penetration attacks in order to meet the security objectives. The protection against attacks with a high attack potential dictates a strength of function rating of “high”. This SoF claim is only applicable to functions in the TOE which are realised using probabilistic or permutational mechanisms.

6.2.2 Fulfilment of TOE objectives by the TOE functional requirements

	SO. EMAN_DESIGN	SO.SELF_TEST	SO.KEY_SECRECY	SO.TAMPER_ID	SO.TAMPER_RESISTANCE	SO.KEY_UPDATE	SO.TAC_CONFIRM	SO.TAC_SECURE	SO.INTEGRITY
FCS_CKM.4			X			X			
FCS_COP.1								X	
FDP_ACC.1/KEY			X			X			X
FDP_ACC.1/TAC							X		X
FDP_ACF.1/KEY			X			X			X
FDP_ACF.1/TAC							X		X
FDP_ITC.1						X			
FDP_RIP.1			X				X		
FDP_SDI.2			X					X	X
FDP_UCT.1						X			
FDP_UIT.1						X			
FIA_AFL.1/PIN							X		
FIA_AFL.1/KEY						X			
FIA_ATD.1							X		
FIA_UAU.1							X		
FIA_UAU.5						X	X		
FIA_UID.1						X	X		
FMT_MSA.1/TAC							X	X	
FMT_MSA.1/KEY						X			
FMT_MSA.2								X	
FMT_MSA.3/TAC							X	X	
FMT_MSA.3/KEY						X			
FMT_MTD.1							X		
FMT_SMF.1/PIN							X		
FMT_SMF.1/KEY						X			
FMT_SMR.1						X	X		
FPT_AMT.1		X						X	
FPT_EMAN.1	X		X						
FPT_FLS.1			X						
FPT_PHP.1				X					
FPT_PHP.3					X				
FPT_TST.1		X							
FTP_ITC.1						X			

SO.EMAN_DESIGN which requires that the TOE is built in such a way as to control the production of intelligible emanations within specified limits is directly fulfilled by the **SFR FPT_EMSEC.1** as this requires that the TOE does not emit intelligible emanations which exceed a certain limit and that it shall not be possible to determine user data of the TOE using these emanations.

SO.SELF_TEST which requires that the TOE has to provide self testing functionality for all security functions is fulfilled by a combination of **FPT_AMT.1** describes that the TOE has to provide a test for the hardware the TOE is relying on and **FPT_TST.1** which describes that the TOE has to be able to run a suite of tests to ensure the correct operation of the TSF.

SO.KEY_SECRECY which describes that the TOE assures the TAC key against attacks is fulfilled by **FCS_CKM.4** which ensures the secure destruction of the keys after an update has been performed, **FDP_ACC.1/KEY** and **FDP_ACF.1/KEY** which specify that nobody is allowed to read out the key, **FDP_RIP.1** which ensures that key in memory which are no longer used are destroyed, **FDP_SDI.2** which specifies the integrity protection of the key and **FPT_FLS.1** which detects insecure states of the TOE. Furthermore **FPT_EMAN.1** contributes to SO.KEY_SECRECY as the design of the TOE which is described in **FPT_EMAN.1** is used to protect the key.

SO.TAMPER_ID which requires that the TOE detects physical tampering directly and completely covered by **FPT_PHP.1**.

SO.TAMPER_RESISTANCE which requires that the TOE has to be resistant against physical tampering is directly and completely covered by **FPT_PHP.3**.

SO.KEY_UPDATE specifies that the TOE has to provide a secure mechanism to update the key. This includes the secure transmission to the TOE, the authentication of the terminal which is sending the key and the secure destruction of old keys.

This objective is fulfilled by a combination of **FCS_CKM.4** which describes the secure key destruction method after the key update has been performed, **FDP_ACC.1/KEY** and **FDP_ACF.1/KEY** which define that only an administrator is allowed to update the keys, **FDP_ITC.1** which defines the import policy for the key update, **FDP_UCT.1** which describes that the keys have to be kept confidential during key update, **FDP_UIT.1** which describes that the TOE has to ensure the integrity of the keys, **FIA_AFL.1/KEY** which ensures that the process of key update is blocked after a certain number of unsuccessful authentication attempts, **FIA_UAU.1** and **FIA_UAU.5** which describe the authentication mechanisms of the terminal, **FIA_UID.1** which requires user identification, **FMT_MSA.1/KEY** which limits the ability to change security attributes for key update to administrators, **FMT_MSA.3/KEY** which defines that nobody is allowed to overwrite the initial values for the security attributes, **FMT_SMF.1/KEY** which defines the management functions for the key update, **FMT_SMR.1** which describes the roles, the TOE has to maintain and **FTP_ITC.1** which describes the requirements for the trusted channel which also include terminal authentication.

SO.TAC_CONFIRM describes that the TOE has to provide a confirmation mechanism which requires the user to confirm the TAC generation. In terms of SFRs this mechanism is modelled as an authentication mechanism as follows:

FDP_ACC.1/TAC and **FDP_ACF.1/TAC** describe the rules for access control related to the TAC generation and the PIN, **FDP_RIP.1** defines that PINs which are no longer used are securely destroyed from memory, **FIA_AFL.1/PIN** defines the authentication failure handling for the TAC generation, **FIA_ATD.1** defines the user attributes which are used for access control, **FIA_UAU.1**, **FIA_UAU.5** and **FIA_UID.1** describe the multiple authentication mechanisms and that each user has to be identified/authenticated before he is allowed to generate the TAC, **FMT_MSA.1/TAC** defines that nobody is allowed to change the security attribute regarding the card holder, **FMT_MTD.1** defines that only the card holder and an administrator are allowed to change the PIN, **FMT_SMF.1/PIN** defines the management function to change the PIN and **FMT_SMR.1** describes the roles, the TOE has to maintain.

SO.TAC_SECURE which requires that the TAC which is generated by the TOE cannot be forged is covered by a combination of **FCS_COP.1** which defines the cryptographic operation to generate the TAC, **FDP_SDI.2** which is used to ensure the integrity of the data which is used to generate the TAC, **FMT_MSA.1/TAC**, **FMT_MSA.3/TAC** and **FMT_MSA.2** which describe the handling of the security attributes which are involved in the TAC generation, **FPT_AMT.1** to ensure the correct operation of the function to generate a TAC.

SO.INTEGRITY which requires that the TOE protects that data in its storage against unauthorized modification is covered by **FDP_ACC.1/KEY** which describes the access control policy for the cryptographic keys together with **FDP_ACF.1/KEY** and **FDP_ACC.1/TAC** which describes the access control policy together with **FDP_ACF.1/TAC** for the TAC. Beside these requirements which are used to decide whether an access attempt to an asset is authorized, **FDP_SDI.2** is used to ensure the integrity of data when stored in the memory of the TOE.

6.2.3 Fulfilment of IT environment objectives by the IT environment functional requirements

	SOE.PERSO	**SOE.KEY**
FCS_CKM.1/ENV		X
FDP_UCT.1/ENV	X	
FDP_UIT.1/ENV	X	
FTP_ITC.1/ENV	X	

Only **SOE.PERSO** and **SOE.KEY** contain requirements for the IT-environment. The requirements for the key out of **SOE.KEY** are directly and completely covered by **FCS_CKM.1/ENV**.

The requirements from **SOE.PERSO** are covered by a combination of **FDP_UCT.1/ENV** which deals with the confidentiality of data and

FDP_UIT.1/ENV and **FTP_ITC.1/ENV** which describe the requirements for the trusted channel.

6.2.4 Mutual support and internal consistency of security requirements

From the details given in this rationale it becomes evident that the functional requirements form an integrated whole and, taken together, are suited to meet all security objectives. Requirements from [CC] part 2 are used to fulfil the security objectives.

The core TOE functionality is represented by the requirements for TAC generation, the handling of the key and the mechanisms for key update. (FCS_CKM.4, FCS_COP.1, FTP_ITC.1)

Furthermore a set of requirements is used to describe the way these functions should be used and who is allowed to uset them (e.g. FDP_ACC.1/KEY)

In the end this PP contains a set of SFRs which deals with the detection and defeating of attacks to the TOE, resp. SFRs which are used to show that the TOE is working correctly (e.g. FPT_PHP.1, FPT_PHP.3, FPT_TST.1)

Therefore it becomes clear that the SFRs in this PP mutually support each other and form a consistent whole.

821 6.2.5 Fulfilment of TOE SFR dependencies

SFR	Dependencies	Dependency fulfilled?
FCS_CKM.4	FDP_ITC.1, FMT_MSA.2	Yes
FCS_COP.1	FDP_ITC.1, FCS_CKM.4, FMT_MSA.2	Yes
FDP_ACC.1/KEY	FDP_ACF.1	Yes
FDP_ACC.1/TAC	FDP_ACF.1	Yes
FDP_ACF.1/KEY	FDP_ACC.1, FMT_MSA.3	Yes
FDP_ACF.1/TAC	FDP_ACC.1, FMT_MSA.3	Yes
FDP_ITC.1	FDP_ACC.1, FMT_MSA.3	Yes
FDP_RIP.1	-	-
FDP_SDI.2	-	-
FDP_UCT.1	FTP_ITC.1, FDP_ACC.1	Yes
FDP_UIT.1	FTP_ITC.1, FDP_ACC.1	Yes
FIA_AFL.1/PIN	FIA_UAU.1	Yes
FIA_AFL.1/KEY	FIA_UAU.1	Yes
FIA_ATD.1	-	-
FIA_UAU.1	FIA_UID.1	Yes
FIA_UAU.5	-	-
FIA_UID.1	-	-
FMT_MSA.1/TAC	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1	Yes
FMT_MSA.1/KEY	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1	Yes
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1, FMT_MSA.1, FMT_SMR.1	Yes
FMT_MSA.3/TAC	FMT_MSA.1, FMT_SMR.1	Yes
FMT_MSA.3/KEY	FMT_MSA.1, FMT_SMR.1	Yes
FMT_MTD.1	FMT_SMF.1, FMT_SMR.1	Yes
FMT_SMF.1/PIN	-	-
FMT_SMF.1/KEY	-	-
FMT_SMR.1	FIA_UID.1	Yes
FPT_AMT.1	-	
FPT_EMAN.1	-	
FPT_FLS.1	ADV_SPM.1	Yes
FPT_PHP.1	-	-
FPT_PHP.3	-	-

FPT_TST.1	FPT_AMT.1	Yes
FTP_ITC.1	-	-

6.2.6 Appropriateness of TOE assurance requirements

The assurance level for this protection profile is EAL4 augmented. EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices.

It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.

The TOE described in this protection profile is just such a product. Augmentation results from the selection of:

AVA_IMP.2 Implementation of the TSF
AVA_VLA.4 Vulnerability Assessment - Vulnerability Analysis – Highly resistant

The main function of the TOE is to protect the cryptographic key which is used to generate the TAC. If an attacker would get knowledge of one or more of these keys, the whole financial system in which the TOE is used may become insecure. Therefore it is reasonable to assume a high attack potential for an attacker and to augment EAL 4 by **AVA_VLA.4.**

AVA_VLA.4 has the following dependencies:

- ADV_FSP.1 Informal functional specification
- ADV_HLD.2 Security enforcing high-level design
- ADV_IMP.1 Subset of the implementation of the TSF
- ADV_LLD.1 Descriptive low-level design
- AGD_ADM.1 Administrator guidance
- AGD_USR.1 User guidance

All of these are met or exceeded in the EAL4 assurance package.

The augmentation by **ADV_IMP.2** requests that the evaluator reviews the complete implementation of the TSF. This is useful as an additional input for AVA_VLA.4 as the evaluation gains knowledge about the complete internal structure of the TOE and is able to use this knowledge for AVA_VLA.4. Therefore it is reasonable to augment EAL4 by **ADV_IMP.2**.

ADV_IMP.2 has the following dependencies:

- ADV_LLD.1 Descriptive low-level design
- ADV_RCR.1 Informal correspondence demonstration
- ALC_TAT.1 Well-defined development tools

All of these are met or exceeded in the EAL4 assurance package.

6.3 Rationale for Extensions

Remarks: Definition of this family is based on the FPT_EMSEC of the SSCD PP [SSCD].

The additional family FPT_EMAN (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 cryptographic keys 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.

6.3.1 FPT_EMAN TOE Emanation

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling:

FPT_EMAN.1 TOE Emanation has two constituents:

- FPT_EMAN.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.
- FPT_EMAN.1.2 Interface Emanation requires not emit interface emanation enabling access to TSF data or user data.

Management: FPT_EMAN.1

There are no management activities foreseen.

Audit: FPT_EMAN.1

There are no actions identified that should be auditable if FAU_GEN Security audit data generation is included in the PP/ST.

6.3.1.1 TOE Emanation (FPT_EMAN.1)

FPT_EMAN.1.1 The TOE shall not emit [*assignment: types of emissions*] in excess of [*assignment: specified limits*] enabling access to secret data including cryptographic keys, especially the TAC key.

FPT_EMAN.1.2 The TSF shall ensure that nobody is able to use *[assignment: types of emissions]* to gain access to secret data including cryptographic keys, especially the TAC key.

Hierarchical to: No other components.

Dependencies: No other components.

905 # 7 Appendix

906 ## 7.1 Abbreviations

907 ### 7.1.1 TOE related abbreviations

Abbreviation	Explanation
AEF	Active Elementary File
APDU	Application Protocol Data Unit
ATM	Automated Teller Machine
CD/ATM	Cash Dispenser/Automated Teller Machine
DF	Dedicated File
DFA	Differential Fault Analysis
DPA	Differential Power Attack
ECB	Electronic Codebook
EEPROM	Electrical Erasable Programmable Read Only Memory
EF	Elementary File
ES	Embedded Software
FISC	Financial Information Services CO., LTD.
ICC	Integrated Circuit Controller
ID	Identification
ITSEC	Information Technology Security Evaluation Criteria
LC	Life Cycle
LRC	Longitudinal Redundancy Check
MF	Master File
NEF	Neutral Elementary File
P-Code	Process Code
PIN	Personal Identification Number
ROM	Read-Only Memory
TAC	Transaction Authentication Code
SPA	Sequential Power Attack
MAC	Message Authentication Code

908 **Table 7: TOE related abbreviations**

909 ### 7.1.2 CC related abbreviations

Abbreviation	Explanation
ST	Security Target
TOE	Target of evaluation
PP	Protection Profile
SFP	Security Function Policy
SF	Security Function
SOE	Security Objectives for the Environment
TSF	TOE Security Function
TSP	TOE Security Policy
NITR	Security requirements for the Non-IT environment

910 **Table 8: CC related abbreviations**

7.2 Glossary

7.3 References

[3DES] Federal Information Processing Standard Publication, FIPS PUB 46-3 October 1999.

[ANSI X9.52] Triple Data Encryption Algorithm Modes of Operation

[ANSI X9.9] Financial Institution Message Authentication

[CC] Common Criteria for information Technology Security evaluation, January 2004, Version 2.2 incorporated with all final comments until April 30th 2005

[CEM] Common Evaluation Methodology for information Technology Security, January 2004, Version 2.2

[SSCD] Secure Signature Creation Device Protection Profile, Type 2, ESIGN Workshop - Expert Group F, Version 1.04, July 2001

[FIPS_A] FIPS PUB 140-2 Annex A: Approved Security Functions, Draft Version, May 19th 2005