Security Target for BAROC/FISC TSAM 1.0

File Name: ST_FISCTSAM_1.0.0

Version: 1.0.0

Date: 2008-05-21

Authors: BAROC & FISC

TOE / TOE Version: BAROC/FISC TSAM 1.0

Table of Contents

List of Figures and Tables

1 ST Introduction

1.1 ST Identification

Title:	Security Target for BAROC/FISC TSAM 1.0
TOE:	BAROC/FISC TSAM 1.0
Guidance:	Administrator and User Guidance for BAROC/FISC TSAM 1.0, version 1.0.0, date: 2008-05-21 BAROC/FISC SHA-1 hash value of the PDF version: 94db00658c87902818433487eed82a88d8408114
Document Version:	1.0.0
Document Date:	2008-05-21
Author:	BAROC & FISC
CC version used:	CC V2.3, CEM V2.3, including all corresponding FIs, as applicable
CC Conformance:	Conformant to CC V2.3 part 2 extended and conformant to CC V2.3 part 3 augmented (EAL4 augmented by ADV_IMP.2 and AVA_VLA.4)
Certification ID:	BSI-DSZ-CC-0442
Evaluation Body:	TÜViT GmbH, Germany
Certification Body:	BSI, Germany

1.2 ST Overview

After a successful chip migration of ATM cards in 2005 for conventional online transactions of cash withdrawal and fund transfer via ATM in Taiwan, FISC would in addition like to promote the debit solution for point of sales (POS) with the chip ATM cards. For this to be done, the confidentiality and integrity of data transfer between a POS terminal and its acquiring bank must be assured as a prerequisite. FISC therefore comes up with the development of TSAM (Terminal Security Access Module), the TOE, to be used by POS terminals to ensure the confidentiality and integrity of data transfer. The TOE is composed of a JavaCard applet (TSAM applet) and NXP P541G072V0P smart card controller (the latter consists of JCP (JavaCard Platform) and SCP (Smart Card Platform)). This security target is for the composite TSAM TOE.

The main objectives of this security target 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 environment.
- To specify the security requirements, which include the TOE security functional requirements as of CC part 2 and the assurance requirements as of CC part 3.
- To setup the TOE summary specification that includes the TOE security function specifications and the assurance measures.

1.3 CC Conformance Claims

This ST is claimed to be conformant with the Common Criteria Version 2.3 ([CC]):

- Security functional requirements are conformant to CC Part 2 extended (extended requirements have been introduced for the underlying platform in [JCOP41V231]).
- Security assurance requirements are conformant to CC Part 3 augmented: EAL4 augmented by AVA_VLA.4 (highly resistant) and ADV_IMP.2 (implementation of the TSF).

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

Concerning the use of random numbers additionally conformance to [AIS20], class K3, SOF-high is claimed ([JCOP41V231ST] already does so for the underlying platform).

This Security Target claims conformance to [JCSPP], Minimal Configuration ([JCOP41V231ST] already does so for the underlying platform).

2 TOE Description

2.1 Overview

TSAM, the TOE, is short for Terminal Security Access Module and, as its name implies, TSAM helps secure transactions in-between POS terminal and the remote host application in a way that it assures integrity, authenticity and confidentiality of POS transactions by encryption, decryption and MAC generation. The functions of TSAM come as follows.

1. TSAM is provisioned with a management key, an encryption key, a decryption key and a MAC generation key.
2. POS terminal is equipped with a TSAM in one of its slots. The terminal asks for data encryption from TSAM when it is submitting a transaction to the remote host. The terminal performs encryption of sensitive part of the transaction message by sending it to TSAM via "Data Encryption by Working Key" command and TSAM responds with encrypted datagram. The terminal can also perform decryption of encrypted part of the received transaction message by sending it to TSAM via "Data Decryption by Working Key" command and TSAM responds with decrypted result.
3. By using TSAM, the terminal calculates MAC for each transaction. The terminal prepares the transaction representation from the transaction message and sends the transaction representation to TSAM via "Generate MAC by Working Key" command. TSAM responds with a MAC over the data it receives from its interface.
4. TSAM is managed by the remote host, which means the management key and working keys (encryption, decryption and MAC) are subject to be changed over time via online transaction. The key management must be secure, and therefore, there is a unique management key for each TSAM so that the remote host can assure the integrity, confidentiality and, of course, authenticity of the key management process.

To sum up, the security relevant functions provided by TSAM help the remote host to assure that every transaction comes from a terminal, equipped with an authentic TSAM, is kept confidential and is not modified.

2.2 TOE Definition

The TOE is composed of a JavaCard applet (TSAM applet) and NXP P541G072V0P smart card controller ([JCOP41V231]), see Figure 2-1. While JCP (JavaCard Platform) resides in ROM, the “TSAM Applet” resides in EEPROM of [JCOP41V231].

[JCOP41V231] has been evaluated before as referred to [JCOP41V231ST] and the respective certification report (BSI-DSZ-CC-0426). The TSAM applet is loaded and installed into [JCOP41V231], and therefore, the TOE is a composition of the TSAM applet and [JCOP41V231]. The GlobalPlatform keys necessary for applet management are not delivered together with the TOE, therefore <u>it will not be possible to delete the TSAM applet from or install additional applets into the smart card controller after delivery</u>.

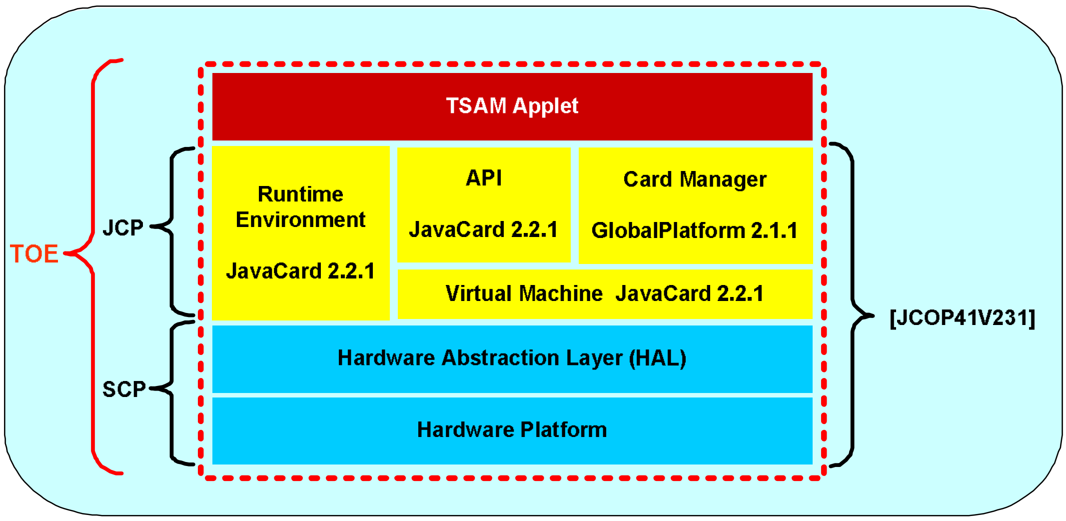

Figure 2-1: The TOE architecture

More information about the structure of JCP (JavaCard Platform) and SCP (Smart Card Platform) can be found in [JCOP41V231ST].

2.3 TOE Boundaries

2.3.1 TOE Physical Scope and Boundary

The physical boundary of the TOE is represented by the surface of [JCOP41V231]. This surface and the embedded physical interface are compliant to ISO 7816 part 2.

While JCP (JavaCard Platform) resides in ROM, the TSAM applet resides in EEPROM of [JCOP41V231].

[JCOP41V231] provides different external interfaces and corresponding protocols. In the TOE only the contact interface and only the corresponding protocol T=1 are available. Contactless interface and USB 2.0 interface implementations of [JCOP41V231] are physically present in the TOE, but are not usable, as these interfaces are not contacted and as the corresponding protocols T=CL, MIFARE and USB protocol are disabled in the TOE (disabled MIFARE part physically present in [JCOP41V231] is not shown in Figure 2-1.hereinabove).

In broadest sense also the guidance documentation can be seen as part of the physical scope of the TOE (see section 1.1 hereinbefore for a detailed reference).

2.3.2 TOE Logical Scope and Boundary

The TOE logical interface is represented by a set of APDU commands which are compliant to ISO 7816 part 4 (augmented with additional commands). At its logical boundary, the TOE provides functions of encryption, decryption, MAC generation and secure key updates.

The TOE provides the following security functionalities:

- encryption, decryption and MAC generation,
- user authentication,
- confidentiality and integrity protection of communication data,
- access control,
- life cycle management,
- stored data integrity protection, and
- increments of serial numbers.

2.4 TOE Life Cycle

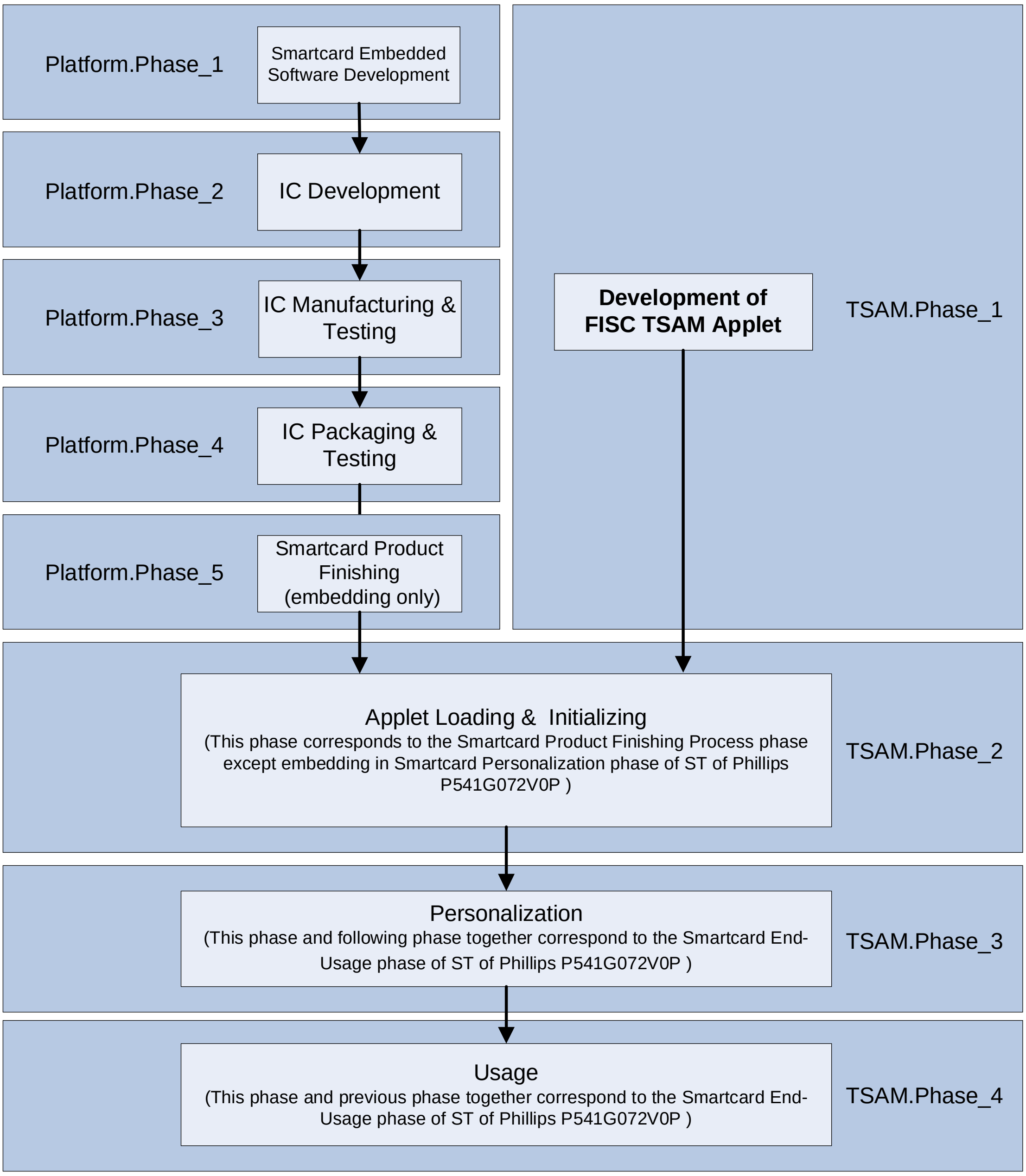

Figure 2-2: TOE life cycle

Figure 2-2 shows development phases and operation phases of the TSAM TOE. Each of the phases is described as follows:

Platform.Phase_1: This phase corresponds to the Smartcard Embedded Software Development phase of [JCOP41V231ST].

Platform.Phase_2: This phase corresponds to the IC Development phase of [JCOP41V231ST].

Platform.Phase_3: This phase corresponds to the IC Manufacturing and Test phase of [JCOP41V231ST].

Platform.Phase_4: This phase corresponds to the IC Packaging and Test of [JCOP41V231ST].

Platform.Phase_5: This phase corresponds to the embedding part of the Smartcard Product Finishing Process phase of [JCOP41V231ST]. Embedded smartcard products, together with the GlobalPlatform keys, are delivered to the TSAM production site in TSAM.Phase_2.

TSAM.Phase_1: This phase consists of the development of the TSAM applet which is to be loaded and installed into [JCOP41V231]. GlobalPlatform keys of [JCOP41V231] are needed for performing the loading and installing of the TSAM applet. TSAM applet is delivered to the TSAM production site in TSAM.Phase_2.

TSAM.Phase_2: This phase, together with Platform.Phase_5, corresponds to the Smartcard Product Finishing Process phase of [JCOP41V231ST]. It consists of the process for loading, installing and initializing the TSAM applet with GlobalPlatform keys. This is done at the production site. After loading and installation of the TSAM applet, the TOE is completed and its security functionality is operative. During subsequent initialization, which is already under control of TSAM's security functionality, the initial management key is written, which is necessary for personalization. The initialized TOE and the corresponding initial management key are delivered to the TSAM issuer site (see TSAM.Phase_3).

TSAM.Phase_3: This phase, together with TSAM.Phase_4, corresponds to the Smartcard End-usage phase of [JCOP41V231ST]. It consists of personalization process of TSAM by the issuer, which includes doing the mandatory first update of the management key and writing of terminal management data. The process is done at TSAM issuer site.

TSAM.Phase_4: This phase, together with TSAM.Phase_3, corresponds to the Smartcard End-usage phase of [JCOP41V231ST]. At this phase, POS terminal uses TSAM for security operations. The TSAM issuer, at this phase, can do update of the management key and writes of working keys.

2.5 Roles

R.Initializer: This is the role that instantiates and initializes the TSAM TOE. This role belongs to the production environment of the TOE, nevertheless, initialization of the management key is already controlled by TOE functionality.

R.Issuer: This is the user who issues TOE and performs management of the management key, working keys and terminal management data.

R.POS_Terminal: This is the device that uses the TOE for data encryption, decryption and MAC generation for POS transactions. The user guidance for this role will be addressed to the developer of the POS terminal.

3 TOE Security Environment

3.1 Assets

The TSAM applet corresponds to D.APP_CODE asset of the ST of NXP P541G072V0P [JCOP41V231ST].

The following assets are corresponding to D.APP_C_DATA (confidential sensitive data of the TSAM applet), D.APP_I_DATA (integrity sensitive data of the TSAM applet) and D.APP_KEYS (cryptographic keys owned by the TSAM applet) of [JCOP41V231ST].

3.1.1 GlobalPlatform Keys (GPKs)

GlobalPlatform keys are 3/DES keys that are used by R.Initializer to protect loading and installing of the TSAM applet by security functionalities of [JCOP41V231].

GPKs also protect initialize of management key (see section 3.1.2 below). This takes place during production; nevertheless, it is already controlled by security functionalities of the TOE.

The TOE has to maintain the integrity and confidentiality of GPKs (this is solely provided by functionality of [JCOP41V231]).

GPKs are not delivered with the TOE.

3.1.2 Management Key (MK)

Management key is a 3/DES key that's used by R.Issuer to protect writes of working keys (see section 3.1.3 below) and key updates of MK itself. It protects key updates and writes in a way that the confidentiality, integrity and authenticity are assured. MK also protects writes of terminal management data (see section 3.1.4 below) of the TOE in a similar way that the integrity and authenticity of the data are assured.

MK is written into the EEPROM of [JCOP41V231] in TSAM.Phase_2. It can be updated in TSAM.Phase_3 and TSAM.Phase_4. The TOE has to maintain the integrity and confidentiality of MK.

3.1.3 Working Keys (WKs)

Working keys are 3/DES keys that are stored in the EEPROM of [JCOP41V231]. There are three WKs in the TOE, which are encryption key, decryption key and MAC generation key. The R.POS_Terminal requests for cryptographic services supported by the TOE to encrypt, decrypt and/or generate MACs over transaction data (see section 3.1.7) by the encryption key, decryption key and/or MAC generation key, respectively.

WKs can be written during TSAM.Phase_4. The TOE has to maintain the integrity and confidentiality of any of the WKs.

3.1.4 Terminal Management Data (TMD)

TMD is composed of a merchant identifier (MID), a terminal identifier (TID), a transaction serial number (TSN) and a batch settlement number (BSN).

TMD is stored in the EEPROM of [JCOP41V231] in TSAM.Phase_3. In TSAM.Phase_4, TMD can be read out of the TOE, and TSN and BSN can be incremented.

The TOE has to ensure the integrity of TMD during writes and storage.

3.1.5 Retry Counter (RC)

This is TSF data which is the counter for accumulative consecutive failure attempts of external authentication with MK. Whenever RC reaches 3, no further attempts of external authentication with MK will be allowed. In this case, there is no way to reset RC. The integrity of RC must be maintained by the TOE.

3.1.6 Life Cycle State (LCS)

This is TSF data which is used to manage life cycle state of the TOE. The life cycle state of the TOE starts from TSAM.Phase_2, changes to TSAM.Phase_3 and finally changes to TSAM.Phase_4 subsequently. The change of the life cycle state is irreversible. The integrity of LCS must be maintained by the TOE.

3.1.7 Transaction Data (TD)

This is user data that the TOE receives from its interface. The data can be subject to encryption, decryption, or MAC generation with the corresponding WK. The data is not stored permanently in the TOE.

3.2 Assumptions (about the environment)

The following set of assumptions incorporates those assumptions made in [JCOP41V231ST], which are still relevant for this composite TOE. Some of the assumptions made in [JCOP41V231ST] are covered by development and production of TSAM and are therefore not listed here. Please see remarks after list of assumptions below and PP claims rationale in section 8.4 hereinafter.

A.DLV *Delivery of TOE and its guidance documents*

It is assumed that R.Issuer and the developer of the R.POS_Terminal verify the hash values of their guidance documents to assure a secure delivery of it. It is also assumed that R.Issuer will only issue the TOE after a correct MAC verification of the delivered management key.

A.USE_DIAG *Use of secure communication protocols*

It is assumed that the environment supports and uses secure communication protocols offered by the TOE.

A.KEYS *Key protection and key quality*

The management key and working keys which are stored and processed outside the TOE during personalization and usage phases are assumed to be protected for confidentiality and integrity.

Cryptographic keys created in the environment to be used within the TOE have to have sufficient quality (e.g. by using a random number generator for key generation).

A.DEV *Development security*

It is assumed that no native codes will be loaded into [JCOP41V231] during development and production phases of the TOE. During development, byte code verification will be performed on the TSAM applet. During production, only TSAM applet will be installed. GlobalPlatform keys are not delivered to R.Issuer and R.POS_Terminal.

It is also assumed that TOE development and test information during TSAM.Phase_1 and TSAM.Phase_2 is protected in a secure environment for its integrity and confidentiality. In case of delivery between different actors like applet developers and applet installers, this information is also protected in the same manner as aforementioned.

Remarks:

- A.DLV covers the assumption A.DLV_PROTECT of [JCOP41V231ST] because the procedures addressed in A.DLV_PROTECT are mostly covered by evaluation of development security, configuration management and delivery for the TOE. A.DLV assumes the remaining responsibilities of the users of the TOE to ensure a complete secure delivery process.
- A.TEST_OPERATE of [JCOP41V231ST] is completely covered by the evaluation of development security, configuration management and delivery for the TOE because the development and production of the TOE covers phases of 4, 5 and 6 of [JCOP41V231ST].
- A.USE_DIAG is a mere re-statement of the assumption A.USE_DIAG of [JCOP41V231ST].
- A.KEYS directly covers the assumption A.USE_KEYS of [JCOP41V231ST] with additional refinements and extensions.
- A.NATIVE of [JCOP41V231ST] is covered by A.DEV because in the scope of the TSAM production and operation no native code will be loaded into the smart card controller.
- A.NO-DELETION and A.NO-INSTALL of [JCOP41V231ST] are covered by A.DEV because the necessary GlobalPlatform keys are not delivered to the R.Issuer and the R.POS_Terminal (or its developer), therefore it will not be possible to delete the TSAM applet from or install additional applets into the smart card controller.
- A.VERIFICATION of [JCOP41V231ST] is covered by A.DEV because byte code verification will be performed during the development/production.

3.3 Threats

This section introduces the threats to the assets against which specific protection within the TOE or its environment is required. It is assumed that all attackers have high level of expertise, opportunity and resources. In [JCOP41V231ST], general threats for smart card native operating systems were defined and supplemented by

Java Card specific threats from [JCSPP] (see section 3.3.2). Additionally in section 3.3.1 hereinafter the TSAM-specific threats are listed.

3.3.1 Threats not contained in [JCOP41V231ST]

T.INTEGRITY *Integrity of security relevant data*

An attacker or memory errors may change MK, WK, TMD, LCS and RC in storage without the TOE being able to detect it, which leads to usage of corrupted data.

T.TMD_ACCESS *Access to terminal management data*

An unauthorized user, other than R.Issuer, may perform writes of TMD. One possibility would be that the unauthorized user records authorized update of TMD during communication and resends it to the TOE (replay attack).

The TMD of an authorized update may be modified during communication but the TOE does not detect the modification.

T.KEY_ACCESS *Access to MK and WK*

An unauthorized user, other than R.Initializer, may perform initialize of MK. An unauthorized user, other than R.Issuer, may perform updates of MK or writes of WKs. One possibility would be that the unauthorized user records authorized initialize and updates of MK or writes of WK during communication and resends them to the TOE (replay attack).

The MK or WKs of an authorized initialize, update or write may be modified during communication but the TOE does not detect the modification.

An attacker may eavesdrop on the MK or WK during communication to get the key value that's being used by the TOE. An attacker or a user may try to read the MK or WKs from the TOE's user visible interfaces. An attacker may also try to gain previous values of MK or WKs from the TOE.

3.3.2 Threats from [JCOP41V231ST]

The following threats have already been regarded during development and manufacturing of [JCOP41V231] as confirmed by the corresponding evaluation.

The threats listed here are just a brief summary. For corresponding explanation and application note, please see [JCOP41V231ST].

Table 3-1 identifies the threats that are found in [JCOP41V231ST]. The Source column of the table indicates the source protection profile, if there is any, in which the corresponding threat is specified. The Life-Cycle column of the table indicates the phases of the TOE life cycle in which the corresponding threat can take place. Detailed explanation of the phases can be found in section 2.4.

Name	Source	Life-Cycle
T.DEV_IC	-	Platform.Phase_2, Platform.Phase_3
T.DEV_NOS	-	Platform.Phase_1

Name	Source	Life-Cycle
T.DEL_IC_NOS	-	Platform.Phase_1, Platform.Phase_2
T.DEL	-	Platform.Phase_4, TSAM.Phase_2
T.ACCESS_DATA	-	TSAM.Phase_3, TSAM.Phase_4
T.OS_OPERATE	-	TSAM.Phase_3, TSAM.Phase_4
T.OS_DECEIVE	-	TSAM.Phase_3, TSAM.Phase_4
T.LEAKAGE	-	TSAM.Phase_3, TSAM.Phase_4
T.FAULT	-	TSAM.Phase_3, TSAM.Phase_4
T.RND	[PP0002]	TSAM.Phase_3, TSAM.Phase_4
T.PHYSICAL	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.CONFID-JCS-CODE	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.CONFID-APPLI-DATA	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.CONFID-JCS-DATA	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.INTEG-APPLI-CODE	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.INTEG-JCS-CODE	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.INTEG-APPLI-DATA	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.INTEG-JCS-DATA	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.SID.1	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.SID.2	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.EXE-CODE.1	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.EXE-CODE.2	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.NATIVE	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4
T.RESOURCES	[JCSPP]	TSAM.Phase_3, TSAM.Phase_4

Table 3-1: Threats from [JCOP41V231ST]

T.DEV_IC

Theft, modification, disclosure of information related to IC development and manufacturing.

T.DEV_NOS

Theft, modification, or disclosure of NOS related information during NOS development.

T.DEL_IC_NOS

Theft, modification, disclosure of information related to IC or NOS during delivery between IC manufacturer and NOS Developer.

T.DEL

Theft, modification, disclosure of information related to TOE during delivery to IC packaging manufacturer or Smart Card manufacturer or personalizer.

T.ACCESS_DATA

Unauthorized access to sensitive information stored in memories in order to disclose or to corrupt the TOE data (TSF and user data).

T.OS_OPERATE

Modification of the correct NOS behavior by unauthorized use of TOE or use of incorrect or unauthorized instructions or commands or sequence of commands, in order to obtain an unauthorized execution of the TOE code.

T.OS_DECEIVE

Modification of the expected TOE configuration by

- o unauthorized loading of code,
- o unauthorized execution of code
- o unauthorized modification of code behavior

T.LEAKAGE

An attacker may exploit information which is leaked from the TOE during usage of the Smart Card in order to disclose the confidential primary assets.

T.FAULT

An attacker may cause a malfunction of TSF or of the Smart Card embedded NOS by applying environmental stress in order to (1) deactivate or modify security features or functions of the TOE or (2) deactivate or modify security functions of the Smart Card embedded NOS. This may be achieved by operating the Smart Card outside the normal operating conditions

T.RND

Deficiency of Random Numbers: An attacker may predict or obtain information about random numbers generated by the TOE for instance because of a lack of entropy of the random numbers provided.

T.PHYSICAL

The attacker discloses or modifies the design of the TOE, its sensitive data (TSF and User Data) or application code or disables security features of the TOE using pure invasive, physical (opposed to logical) attacks on the hardware part of the TOE.

T.CONFID-JCS-CODE

The attacker executes an application without authorization to disclose the Java Card System code.

T.CONFID-APPLI-DATA

The attacker executes an application without authorization to disclose data belonging to another application.

T.CONFID-JCS-DATA

The attacker executes an application without authorization to disclose data belonging to the Java Card System.

T.INTEG-APPLI-CODE

The attacker executes an application to alter (part of) its own or another application's code.

T.INTEG-JCS-CODE

The attacker executes an application to alter (part of) the Java Card System code.

T.INTEG-APPLI-DATA

The attacker executes an application to alter (part of) another application's data.

T.INTEG-JCS-DATA

The attacker executes an application to alter (part of) Java Card System or API data.

T.SID.1

An applet impersonates another application, or even the JCRE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal.

T.SID.2

The attacker modifies the identity of the privileged roles.

T.EXE-CODE.1

An applet performs an unauthorized execution of a method.

T.EXE-CODE.2

An applet performs an unauthorized execution of a method fragment or arbitrary data.

T.NATIVE

An applet tries to execute a native method to bypass some security function such as the firewall.

T.RESOURCES

An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM.

3.4 Organisational Security Policies (OSP)

3.4.1 OSPs not contained in [JCOP41V231ST]

OSP.TXN_SECURE

The TOE has to provide a function to encrypt, decrypt or generate a MAC over TD with the corresponding WK using 3/DES. No external authentication against the TOE is necessary before the TOE's performing such function.

OSP.SN

The TOE has to provide a function to increment TSN and/or BSN of TMD. The increment of TSN and/or BSN shall not exceed specified limits. No external authentication against the TOE is necessary before the TOE's performing such function.

3.4.2 OSPs from [JCOP41V231ST]

The following OSP has already been regarded during development and manufacturing of [JCOP41V231] as confirmed by the corresponding evaluation.

OSP.IC_ORG

Procedures dealing with physical, personnel, organizational, technical measures for the confidentiality and integrity, of Smart Card Native Operating System (e.g. source code mask and any associated documents) and IC Manufacturer proprietary information (tools, software, documentation, dies ...) shall exist and be applied in IC development and manufacturing .

Procedures shall also ensure the confidentiality and integrity and information during exchange with the NOS developer.

4 Security Objectives

4.1 Security Objectives for the TOE

4.1.1 Security Objectives not contained in [JCOP41V231ST]

SO.KEY_ACCESS *Secure access to MK and WKs*

The TOE has to provide a secure mechanism for R.Initializer to initialize MK. The TOE has to provide a secure mechanism for R.Issuer to perform updates of MK and writes of WKs. This includes mechanisms to ensure the confidentiality and integrity of the keys transferred to the TOE as well as the authentication of R.Initializer or R.Issuer who sends the keys.

Nobody shall be able to read out the MK and WKs. The TOE shall provide safe destruction techniques for the cryptographic keys in case of key updates.

SO.TMD_ACCESS *Secure access to TMD*

The TOE has to provide a secure mechanism for R.Issuer to write TMD. This includes mechanisms to ensure the integrity of the TMD transferred to the TOE as well as the authentication of R.Issuer who sends the TMD.

SO.REPLAY *Replay protection in key access and TMD access*

The TOE has to provide a secure mechanism to assure the same command data used in MK initialize and update, WK write and TMD write cannot be used successfully at the second time.

SO.TXN_SECURE *Cryptographic algorithm security for TD*

On request of R.POS_Terminal, the TOE uses 3/DES to encrypt, decrypt or generate MAC over TD with the corresponding WK.

SO.SN *Increments of TSN and BSN*

On request of R.POS_Terminal, the TOE increments TSN and/or BSN of TMD without exceeding specified limits.

SO.INTEGRITY *Integrity error detection*

The TOE protects RC, LCS and TMD in its storage against undetected modifications by an attacker or due to memory errors. On detection of integrity errors, the following actions shall be performed:

- Prohibit the use of the altered data.
- Inform the user about integrity errors.

Remark:

Integrity protection for MK and WKs is provided by [JCOP41V231] already, as its key objects holding MK and WKs are integrity-protected (this is reflected by O.PROTECT_DATA of [JCOP41V231ST], see following section).

4.1.2 Security Objectives from [JCOP41V231ST]

The following security objectives have already been regarded during development and manufacturing of [JCOP41V231] as confirmed by the corresponding evaluation.

The security objectives listed here are just a brief summary. For corresponding explanation and application note, please see [JCOP41V231ST].

Table 4-1 identifies the security objectives that are found in [JCOP41V231ST]. The Source column of the table indicates the source protection profile, if there is any, in which the corresponding security objective is specified.

Name	Source
O.PROTECT_DATA	-
O.SIDE_CHANNEL	-
O.OS_DECEIVE	-
O.FAULT_PROTECT	-
O.PHYSICAL	-
O.RND	[PP0002]
O.SID	[JCSPP]
O.OPERATE	[JCSPP]
O.RESOURCES	[JCSPP]
O.FIREWALL	[JCSPP]
O.NATIVE	[JCSPP]
O.REALLOCATION	[JCSPP]

Name	Source
O.SHRD_VAR_CONFID	[JCSPP]
O.SHRD_VAR_INTEG	[JCSPP]
O.ALARM	[JCSPP]
O.TRANSACTION	[JCSPP]
O.CIPHER	[JCSPP]
O.PIN-MNGT	[JCSPP]
O.KEY-MNGT	[JCSPP]
O.CARD-MANAGEMENT	[JCSPP]
O.SCP.RECOVERY	[JCSPP]
O.SCP.SUPPORT	[JCSPP]
O.SCP.IC	[JCSPP]

Table 4-1: Security objectives from [JCOP41V231ST]

O.PROTECT_DATA

The TOE shall ensure that sensitive information stored in memories is protected against unauthorized disclosure and any corruption or unauthorized modification. Moreover, the TOE shall ensure that sensitive information stored in memories is protected against unauthorized access. The TOE has to provide appropriate security mechanisms to avoid fraudulent access to any sensitive data, such as passwords, cryptographic keys or authentication data.

O.SIDE_CHANNEL

The TOE must provide protection against disclosure of primary assets including confidential data (User Data or TSF data) stored and/or processed in the Smart Card IC by measurement and analysis of the shape and amplitude or by measurement and analysis of the time between events found by measuring signals (for example on the power, clock, or I/O lines).

O.OS_DECEIVE

The TOE must guarantee that only secure values are used for its management and operations, especially system flags or cryptographic assets.

Moreover, the integrity of the whole TOE including the NOS must be guaranteed to prevent disclosing/bypassing of the NOS mechanisms or modifying the expected NOS behavior (for instance, unauthorized code patch, or rewriting).

O.FAULT_PROTECT

The TOE must ensure its correct operation even outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include voltage, clock frequency, temperature, or external energy fields that can be applied on all interfaces of the TOE (physical or electrical).

O.PHYSICAL

The TOE hardware provides the following protection against physical manipulation of the IC, and prevent reverse-engineering (understanding the design and its properties and functions), physical access to the IC active surface (probing) allowing unauthorized memory content disclosure, manipulation of the hardware security parts (e.g. sensors, cryptographic engine or RNG) or manipulation of the IC, including the embedded NOS and its application data (e.g. lock and life cycle status, authentication flags, etc.).

O.RND

The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy.

The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

O.SID

The TOE shall uniquely identify every subject (applet, or package) before granting him access to any service.

O.OPERATE

The TOE must ensure continued correct operation of its security functions. Especially, the TOE must prevent the unauthorized use of TOE or use of incorrect or unauthorized instructions or commands or sequence of commands.

O.RESOURCES

The TOE shall control the availability of resources for the applications.

O.FIREWALL

The TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and the TSFs.

O.NATIVE

The only means that the JCVM shall provide for an application to execute native code is the invocation of a method of the Java Card API, or any additional API.

O.REALLOCATION

The TOE shall ensure that the re-allocation of a memory block for the runtime areas of the JCVM does not disclose any information that was previously stored in that block.

O.SHRD_VAR_CONFID

The TOE shall ensure that any data container that is shared by all applications is always cleaned after the execution of an application. Examples of such shared containers are the APDU buffer, the byte array used for the invocation of the process method of the selected applet, or any public global variable exported by the API.

O.SHRD_VAR_INTEG

The TOE shall ensure that only the currently selected application may grant write access to a data memory area that is shared by all applications, like the APDU buffer, the byte array used for the invocation of the process method of the selected applet, or any public global variable exported by the API. Even though the memory area is shared by all applications, the TOE shall restrict the possibility of getting a reference to such memory area to the application that has been selected for execution. The selected application may decide to temporarily hand over the reference to other applications at its own risk, but the TOE shall prevent those applications from storing the reference as part of their persistent states.

O.ALARM

The TOE shall provide appropriate feedback information upon detection of a potential security violation.

O.TRANSACTION

The TOE must provide a means to execute a set of operations atomically.

O.CIPHER

The TOE shall provide a means to cipher sensitive data for applications in a secure way. In particular, the TOE must support cryptographic algorithms consistent with cryptographic usage policies and standards.

O.PIN-MNGT

The TOE shall provide a means to securely manage PIN objects.

O.KEY-MNGT

The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys.

O.CARD-MANAGEMENT

The card manager shall control the access to card management functions such as the installation, update or deletion of applets. It shall also implement the card issuer’s policy on the card.

O.SCP.RECOVERY

If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state.

O.SCP.SUPPORT

The SCP shall provide functionalities that support the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs. In addition, the smart card platform should also provide basic services which are required by the runtime environment to implement security mechanisms such as atomic transactions, management of persistent and transient objects and cryptographic functions. These mechanisms are likely to be used by security functions implementing the security requirements defined for the TOE.

O.SCP.IC

The SCP shall possess IC security features.

4.2 Security Objectives for the Environment

4.2.1 Security Objectives for The Environment not contained [JCOP41V231ST]

SOE.DLV

R.Issuer and the developer of the R.POS_Terminal shall verify the hash values of their guidance documents as stated in the ST introduction to assure a secure delivery of it. R.Issuer shall only issue the TOE after he could successfully verify the MAC returned by the TOE during first update of MK with the delivered management key.

SOE.USE_DIAG

The environment shall support and use secure communication protocols offered by the TOE.

SOE.KEYS

The management key and working keys which are stored and processed outside the TOE during personalization and usage phases shall be protected for confidentiality and integrity.

Cryptographic keys created in the environment to be used within the TOE have to have sufficient quality by using a random number generator for key generation.

SOE.DEV

No native codes shall be loaded into [JCOP41V231] during development and production phases of the TOE. During development, byte code verification shall be performed on the TSAM applet. During production, only TSAM applet shall be installed. GlobalPlatform keys shall be not delivered to R.Issuer and R.POS_Terminal.

TOE development and test information during TSAM.Phase_1 and TSAM.Phase_2 shall be protected in a secure environment for its integrity and confidentiality. In case of delivery between different actors like applet developers and applet installers, this information shall be also protected in the same manner as aforementioned.

4.2.2 Security Objectives for the Environment from [JCOP41V231ST]

The following security objectives for the environment either have been regarded during development and manufacturing of [JCOP41V231] as confirmed by the corresponding evaluation or are covered by objectives in section 4.2.1.

Table 4-2 identifies the initial security objectives for the environment from [JCOP41V231ST]. For the complete details, please refer to [JCOP41V231ST].

Name	Source	Regards to	Remark
OE.DEV_NOS	-	Platform.Phase_1	Regarded by platform evaluation
OE.DEL_NOS	-	Platform.Phase_1	Regarded by platform evaluation
OE.IC_ORG	-	Platform.Phase_2 Platform.Phase_3	Regarded by platform evaluation
OE.DLV_PROTECT	-	Platform.Phase_3 Platform.Phase_4 TSAM.Phase_2 TSAM.Phase_3 TSAM.Phase_4	Covered by SOE.DLV
OE.DLV_DATA	-	Platform.Phase_4 TSAM.Phase_2	Covered by SOE.DEV
OE.TEST_OPERATE	-	Platform.Phase_4 TSAM.Phase_2	Covered by SOE.DEV
OE.USE_DIAG	-	TSAM.Phase_3 TSAM.Phase_4	Covered by SOE.USE_DIAG
OE.USE_KEYS	-	TSAM.Phase_3 TSAM.Phase_4	Covered by SOE.KEYS
OE.NATIVE	[JCSPP]	TSAM_Phase_2	Covered by SOE.DEV
OE.NO-DELETION	[JCSPP]	TSAM_Phase_2	Covered by SOE.DEV
OE.NO-INSTALL	[JCSPP]	TSAM_Phase_2	Covered by SOE.DEV
OE.VERIFICATION	[JCSPP]	TSAM_Phase_1	Covered by SOE.DEV

Table 4-2: Security objectives for the environment from [JCOP41V231ST]

5 Security Requirements

The minimum strength of function level for the TOE is claimed to be SOF-high. For random number usage conformance to [AIS20] class K3, SOF-high is claimed.

5.1 TOE Security Functional Requirements not contained in [JCOP41V231ST]

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic key destruction (FCS_CKM.4/TSAM)

FCS_CKM.4.1/TSAM The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*previous MK and WKs are physically overwritten by new keys*] that meets the following: [*none*].

5.1.1.2 Cryptographic operation (FCS_COP.1/TSAM)

FCS_COP.1.1/TSAM The TSF shall perform [*encryption, decryption, MAC generation for TD with dedicated keys in TSAM.Phase.4*] in accordance with a specified cryptographic algorithm [*3/DES in ECB or CBC mode*] and cryptographic key sizes [*112 bits*] that meet the following: [*ANSI X 9.52 TECB for encryption/decryption, ANSI X 9.9 with ANSI X 9.52 TCBC Encryption for MAC generation*].

5.1.2 User data protection (FDP)

5.1.2.1 Subset access control (FDP_ACC.1/KEY and FDP_ACC.1/TMD)

FDP_ACC.1.1/KEY The TSF shall enforce the [*Key Access SFP*] on [*subjects: users, objects: MK, WKs and operation: initialize, first update, update, write, read and use*].

FDP_ACC.1.1/TMD The TSF shall enforce the [*TMD Access SFP*] on [*subjects: users, objects: TMD and operation: read, write and increment*].

Application Note:

The operation "use" is applicable to WKs. It means encryption, decryption or MAC generation with the corresponding WK. The operation "increment" is applicable to TSN or BSN of TMD.

5.1.2.2 Security attribute based access control (FDP_ACF.1/KEY and FDP_ACF.1/TMD)

FDP_ACF.1.1/KEY The TSF shall enforce the [*Key Access SFP*] to objects based on the following: [*subject attribute: user role {R.Initializer, R.Issuer, R.POS_Terminal} and object attribute: life cycle state { TSAM.Phase_2, TSAM.Phase_3, TSAM.Phase_4}*].

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

1. *A user with user role {R.Initializer} is allowed to initialize the MK if the life cycle state is {TSAM.Phase_2}.*
2. *A user with user role {R.Issuer} is allowed to do first update of the MK if the life cycle state is {TSAM.Phase_3}.*
3. *A user with user role {R.Issuer} is allowed to do updates of the MK if the life cycle state is {TSAM.Phase_4}.*
4. *A user with user role {R.Issuer} is allowed to do writes of the WK if the life cycle state is {TSAM.Phase_4}.*

5. *A user with user role {R.POS_Terminal} is allowed to use the WK if the life cycle state is {TSAM.Phase_4}.*

]

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 [*rule that no user can read any of the MK and WKs out of the TOE*].

FDP_ACF.1.1/TMD The TSF shall enforce the [*TMD Access SFP*] to objects based on the following: [*subject attribute: user role {R.Issuer, R.POS_Terminal} and object attribute: life cycle state {TSAM.Phase_3, TSAM.Phase_4}*].

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

1. *A user with user role {R.Issuer} is allowed to write TMD if the life cycle state is {TSAM.Phase_3}.*
2. *A user with user role {R.POS_Terminal} is allowed to read TMD and increment TSN/BSN of TMD if the life cycle state is {TSAM.Phase_4}.*

]

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

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

1. *Increment of TSN is denied if the value of TSN is equal to 999999.*
2. *Increment of BSN is denied if the value of BSN is equal to 9999.*

].

Application Note:

R.Initializer and R.Issuer need to be authenticated. R.POS_Terminal doesn't need authentication, i.e., it is an anonymous user.

5.1.2.3 Import of user data without security attributes (FDP_ITC.1/KEY and FDP_ITC.1/TMD)

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

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

FDP_ITC.1.3/KEY The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [

1. *After import of MK by initialize operation, the security attribute life cycle state shall change from TSAM.Phase_2 to TSAM.Phase_3.*

]

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

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

FDP_ITC.1.3/TMD The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [

1. *After import of TMD by write operation, the security attribute life cycle state shall change from TSAM.Phase_3 to TSAM.Phase_4.*

]

5.1.2.4 Stored data integrity monitoring and action (FDP_SDI.2/TSAM)

FDP_SDI.2.1/TSAM The TSF shall monitor user data stored within the TSC for [*integrity errors*] on all objects, based on the following attributes [*checksum for TMD, LCS and RC*].

FDP_SDI.2.2/TSAM Upon detection of a data integrity error, the TSF shall [*inform the user and perform the actions in* Table 5-1 *depending on which object is incurred in the data integrity error*].

Object	Action
RC	No more usage of the MK is allowed (e.g. for authentication).
LCS	Stop operation of the TOE.
TMD	No more read or increment of TMD is allowed.

Table 5-1: Actions on detection of integrity errors

Application Note:

The integrity status for application keys (MK and WKs) are maintained by [JCOP41V231], which monitors integrity when application keys are accessed and stops operation immediately when detecting a corresponding integrity error (therefore preventing that corrupted MK or WKs can be used).

5.1.2.5 Basic data exchange confidentiality (FDP_UCT.1/KEY)

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

Application Note:

This SFR applies to initialize and/or updates of MK and writes of WKs.

5.1.2.6 Data exchange integrity (FDP_UIT.1/TSAM)

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

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

Application Note:

The TOE can detect modification, insertion or replay, but it is not able to distinguish between them. Concerning Key Access SFP, this SFR applies to initializes and/or updates of MK and writes of WKs. Concerning TMD Access SFP, this SFR applies to writes of TMD.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1/TSAM)

FIA_AFL.1.1/TSAM The TSF shall detect when [three consecutive] unsuccessful authentication attempts occur related to [*authentication with MK*].

FIA_AFL.1.2/TSAM When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*no longer allow authentication with MK*].

5.1.3.2 Timing of authentication (FIA_UAU.1/TSAM)

FIA_UAU.1.1/TSAM The TSF shall allow [*encryption, decryption and MAC generation by corresponding WK, reading TMD, incrementing TSN and/or BSN of TMD*] on behalf of the user to be performed before the user is authenticated.

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

5.1.3.3 Single-use authentication mechanisms (FIA_UAU.4/TSAM)

FIA_UAU.4.1/TSAM The TSF shall prevent reuse of authentication data related to [*GlobalPlatform card manager authentication, authentication with MK*].

5.1.3.4 Multiple authentication mechanisms (FIA_UAU.5/TSAM)

FIA_UAU.5.1/TSAM The TSF shall provide [*GlobalPlatform card manager authentication, authentication with MK*] to support user authentication.

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

1. *GlobalPlatform card manager authentication is used for authentication of R.Initializer in TSAM.Phase_2.*
2. *Authentication with MK is used for authentication of R.Issuer in TSAM.Phase_3 and TSAM.Phase_4*

].

Application Note:

Although GlobalPlatform card manager authentication and authentication with MK are both based on 3/DES-based challenge-response protocols, FIA_UAU.5 was chosen for the following three reasons:

1. to explicitly require which authentication mechanism (i.e. based on which key) shall be used for authentication of which user,
2. because the two authentication mechanisms use two different dedicated external interfaces of TSAM,
3. because the two authentication mechanisms differ in their realization: whereas for authentication of R.Initializer internally card manager authentication (SF.I&A) of [JCOP41V231] according to FIA_UAU.1 of [JCOP41V231ST] is used, authentication of R.Issuer using MK is solely implemented in TSAM applet (only using cryptographic primitives of the platform).

5.1.3.5 Timing of identification (FIA_UID.1)

FIA_UID.1.1/TSAM The TSF shall allow [*encryption, decryption and MAC generation by corresponding WK, reading TMD, incrementing TSN and/or BSN of TMD*] on behalf of the user to be performed before the user is identified.

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

5.1.4 Security management (FMT)

5.1.4.1 Management of security attributes (FMT_MSA.1/TSAM)

FMT_MSA.1.1/TSAM The TSF shall enforce the [*Key Access SFP and TMD Access SFP*] to restrict the ability to [modify] the security attributes [*life cycle state*] to [*R.Initializer and R.Issuer*].

5.1.4.2 Secure security attributes (FMT_MSA.2/TSAM)

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

5.1.4.3 Static attribute initialisation (FMT_MSA.3/TSAM)

FMT_MSA.3.1/TSAM The TSF shall enforce the [*Key Access SFP and TMD Access SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

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

Application Note:

The TSAM TOE operates on one fixed set of objects and can not create additional ones. Therefore, the requirement above is only about the initialization of the security attribute life cycle state. “Restrictive” corresponds to a setting to TSAM.Phase_2, which only allows access by R.Initializer.

5.1.4.4 Specification of Management Functions (FMT_SMF.1/TSAM)

FMT_SMF.1.1/TSAM The TSF shall be capable of performing the following security management functions: [*modification of the life state according to FMT_MSA.1.1/TSAM, FDP_ITC.1.3 /KEY and FDP_ITC.1.3 /TMD*].

5.1.4.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1/TSAM The TSF shall maintain the roles [*R.Initializer, R.Issuer and R.POS_Terminal*].

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

Application Note:

R.Initializer and R.Issuer need to be authenticated. R.POS_Terminal doesn’t need authentication, i.e., it is an anonymous user.

5.1.5 Trusted path/channels (FTP)

5.1.5.1 Inter-TSF trusted channel (FTP_ITC.1/TSAM)

FTP_ITC.1.1/TSAM 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/TSAM The TSF shall permit [the remote trusted IT product] to initiate communication via the trusted channel.

FTP_ITC.1.3/TSAM The TSF shall initiate communication via the trusted channel for [*performing initialize, first update, updates and writes of MK, WKs and TMD, as applicable*].

5.2 TOE Security Functional Requirements from [JCOP41V231ST]

In the following table the TOE security functional requirements from [JCOP41V231] are referenced. For details, please refer to [JCOP41V231ST] section 5.1.

Functional Class	Functional Components	
FAU: Security audit	FAU_ARP.1	Security alarms
	FAU_SAA.1	Potential violation analysis
FCS: Cryptographic support	FCS_CKM.1	Cryptographic key generation
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.3	Cryptographic key access
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1	Cryptographic operation
	FCS_RND.1	Quality metric for random numbers
FDP: User data protection	FDP_ACC.1	Subset access control
	FDP_ACC.2	Complete access control
	FDP_ACF.1	Security attribute based access control
	FDP_ETC.1	Export of user data without security attributes
	FDP_IFC.1	Subset Information flow control
	FDP_IFF.1	Simple security attributes
	FDP_ITC.1	Import of user data without security attributes
	FDP_RIP.1	Subset residual information protection
	FDP_ROL.1	Basic rollback
	FDP_SDI.2	Stored data integrity monitoring and action
FIA: Identification and authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_UAU.1	Timing of authentication
	FIA_UAU.3	Unforgeable authentication
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UID.1	Timing of identification
	FIA_UID.2	User identification before any action
	FIA_USB.1	User-subject binding
FMT: Security management	FMT_LIM.1	Limited capabilities
	FMT_LIM.2	Limited availability
	FMT_MSA.1	Management of security attributes
	FMT_MSA.2	Secure security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_MTD.1	Management of TSF data

Functional Class	Functional Components	
	FMT_MTD.3	Secure TSF data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security roles
FPR: Privacy	FPR_UNO.1	Unobservability
FPT: Protection of the TSF	FPT_AMT.1	Abstract machine testing
	FPT_EMSEC.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_RVM.1	Reference mediation
	FPT_SEP.1	TSF domain separation
	FPT_RCV.3	Trusted Recovery
	FPT_RCV.4	Trusted Recovery
	FPT_TDC.1	Inter-TSF basic TSF data consistency
	FPT_TST.1	TSF testing
FRU: Resource utilization	FRU_FLT.2	Limited fault tolerance
FTP: Trusted path/channels	FTP_ITC.1	Inter-TSF trusted channel

979 **Table 5-2: TOE SFRs from [JCOP41V231ST]**

5.3 TOE Security Assurance Requirements

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

Assurance Class	Assurance Components	
ACM: Configuration management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
ADO: Delivery and operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.2	Fully defined external interfaces
	ADV_HLD.2	Security enforcing high-level design
	ADV_IMP.2	Implementation of the TSF
	ADV_LLD.1	Descriptive low-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
ALC: Life cycle support	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
ATE: Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.2	Validation of analysis
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.4	Highly resistant

Table 5-3: Evaluation Assurance Requirements

Remark: [JCOP41V231] has been evaluated according to EAL4 augmented by AVA_VLA.4, ADV_IMP.2, ALC_DVS.2 and AVA_MSU.3. For TSAM TOE evaluation, the same set of security assurance requirements is used except ALC_DVS.2 and AVA_MSU.3. ALC_DVS.1 and AVA_MSU.2 are taken in this evaluation instead of ALC_DVS.2 and AVA_MSU.3, respectively.

5.4 IT Environment Security Requirements not contained in [JCOP41V231ST]

In this section the term “TSF” inside SFRs has been refined to “environment” for clarification. Furthermore the term “a remote trusted IT product” has been refined to “the TOE”. Please note that the dependencies of the following SFRs for the IT environment have not been considered.

5.4.1 Cryptographic key generation

5.4.1.1 Cryptographic key generation (FCS_CKM.1/ENV)

FCS_CKM.1.1/ENV	The environment shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*random number generation*] and specified cryptographic key sizes [*112 bits*] that meet the following: [*none*].

Application Note:

FDP_CKM.1/ENV refers to production of the TOE as well as to usage after TOE delivery.

5.4.2 User data protection

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

FDP_UCT.1.1/ENV	The environment shall enforce the [*Key Access SFP*] to be able to [transmit] objects in a manner protected from unauthorised disclosure.

5.4.2.2 Data exchange integrity (FDP_UIT.1/ENV)

FDP_UIT.1.1/ENV	The environment shall enforce the [*Key Access SFP and TMD Access SFP*] to be able to [transmit] user data in a manner protected from [modification, insertion, replay] errors.
FDP_UIT.1.2/ENV	The environment shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion, replay] has occurred.

Application Note:

FDP_UCT.1/ENV and FDP_UIT.1/ENV refer to production of the TOE as well as to usage after TOE delivery. In both phases the environment is only transmitting user data to the TOE, therefore FDP_UIT.1.2/ENV is not applicable.

5.4.3 Trusted path/channels

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

FTP_ITC.1.1/ENV	The environment shall provide a communication channel between itself and the TOE 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 environment shall permit [the environment] to initiate communication via the trusted channel.
FTP_ITC.1.3/ENV	The environment shall initiate communication via the trusted channel for [*loading of D.App_Code, setting the Card Life Cycle State, initializing MK during TSAM.Phase_2 and first update of MK during TSAM.Phase_3*].

Application Note:

Concerning loading of D.App_Code, setting the Card Life Cycle State and initializing MK FTP_ITC.1/ENV refers to production of the TOE, concerning first update of MK FTP_ITC.1/ENV refers to usage after TOE delivery. FTP_ITC.1.1/ENV was already included in [JCOP41V231ST] as an SFR for the IT environment, but there its scope was limited to loading of D.App_Code and setting the Card Life Cycle State.

5.5 IT Environment Security Requirements from [JCOP41V231ST]

In the following table the security functional requirements for the IT environment from [JCOP41V231] (concerning byte code verification) are referenced. For details, please refer to [JCOP41V231ST] section 5.3.1. These requirements refer solely to production of the TOE, not to usage after TOE delivery. Please note that (1) FTP_ITC.1/ENV, which is also defined as an SFR for the IT environment in [JCOP41V231ST], is listed in section 5.4 above as its scope has been extended compared to [JCOP41V231ST], and (2) that all SFRs in the table below except FMT_SMF.1/BCV are defined by [JCSPP].

Functional Class	Functional Components	
FDP: User Data Protection	FDP_IFC.2/BCV	Complete information flow control
	FDP_IFF.2/BCV	Hierarchical security attributes
FMT: Security Management	FMT_MSA.1/BCV	Management of security attributes
	FMT_MSA.2/BCV	Secure security attributes
	FMT_MSA.3/BCV	Static attribute initialization
	FMT_SMF.1/BCV	Specification of Management Functions
	FMT_SMR.1/BCV	Security roles
FRU: Resource Utilization	FRU_RSA.1/BCV	Maximum Quotas

Table 5-4: IT Environment SFRs from [JCOP41V231ST]

6 TOE Summary Specification

6.1 Security Functions

SF.AUT_GP *TSAM_GlobalPlatform authentication*

SF.AUT_GP will authenticate the user by a challenge-response mechanism using GlobalPlatform keys. For each authentication attempt, SF.AUT_GP will present a new random number[1] as a challenge. Only if the user provides the corresponding correct response, the user is authenticated as the initializer (R.Initializer). In case of a successful authentication, SF.AUT_GP will establish session keys that are later on used by SF.CP_GP.

SF.AUT_GP is only available in TSAM.Phase_2.

Remark: This security function has to be used by the initializer (R.Initializer) before the initializer being able to initialize MK in TSAM.Phase_2. The initializer belongs to the production environment of the TOE, nevertheless, MK initialize is already access controlled by TOE functionality.

SF.CP_GP *TSAM_GlobalPlatform communication protection*

SF.CP_GP provides confidentiality and integrity protection of communication data between the user and the TOE. This is done by decryption and verification of cryptographic checksum using session keys. The corresponding session keys are established after a successful authentication by SF.AUT_GP.

SF.CP_GP is only available in TSAM.Phase_2.

Remark: This security function is used by the initializer (R.Initializer) to protect the transfer of MK while initializing it in TSAM.Phase_2. The initializer belongs to the production environment of the TOE, nevertheless, MK initialize is already access controlled by TOE functionality.

SF.CP_MK *Communication protection with MK*

SF.CP_MK assures integrity, authenticity and optionally confidentiality of communication data between the user and the TOE for a single command. This is done by MAC verification and decryption using session keys which are only valid for this command. To do so, SF.CP_MK performs the following five steps:

1. For establishing the session keys, a random number RN[1] is provided by SF.CP_MK to the user as the very first step.
2. SF.CP_MK receives the command from the user.
3. SF.CP_MK checks the value of RC. If it is equal to 3, SF.CP_MK returns an error code and stops processing. Otherwise, it continues with the next step.

---

[1] Used random numbers are taken from [JCOP41V231], which implements a random number generator conformant to [AIS20], class K3, SOF-high (see SF.EmbeddedSoftware).

4. SF.CP_MK generates the session key for MAC verification by encrypting RN with MK. SF.CP_MK verifies the MAC within the command. If verification fails, it increases RC, returns an error code and stops processing. Otherwise, the issuer (R.Issuer) is authenticated, and SF.CP_MK resets RC to zero and continues with the next step.
5. If the command includes encrypted data, SF.CP_MK generates the session key for decryption by encrypting the inverse of RN with MK and SF.CP_MK decrypts the encrypted data.

SF.CP_MK is only available in TSAM.Phase_3 and TSAM.Phase_4.

Remark 1: This security function is used to protect confidentiality and integrity of the MK during first update. It is also used to protect confidentiality and integrity of MK and WKs during updates and writes, respectively. This security function protects integrity of TMD during writes.

Remark 2: The MAC verification assures authentication of the issuer as well as integrity of the communication data.

SF.AC *Access control*

SF.AC enforces access control rules based on commands, user roles and life cycle state. For commands needing authentication, SF.AC identifies user roles R.Initializer and R.Issuer with SF.AUT_GP and SF.CP_MK, respectively. For commands not needing authentication, SF.AC identifies the user role as R.POS_Terminal.

The following is SF.AC-enforced access control rules:

1. The initializer (R.Initializer) is allowed to initialize MK in TSAM.Phase_2.
2. The issuer (R.Issuer) is allowed to perform first update of MK in TSAM.Phase_3. The issuer is also allowed to perform updates of MK and writes of WKs in TSAM.Phase_4.
3. No user can read any of the MK and WKs out of the TOE.
4. The issuer (R.Issuer) is allowed to write TMD in TSAM.Phase_3.
5. The user R.POS_Terminal is allowed to read TMD out of the TOE in TSAM.Phase_4.
6. The user R.POS_Terminal is allowed to increment TSN of TMD in TSAM.Phase_4 unless the value of TSN is equal to 999999.
7. The user R.POS_Terminal is allowed to increment BSN of TMD in TSAM.Phase_4 unless the value of BSN is equal to 9999.
8. The user R.POS_Terminal is allowed to use WKs according to SF.USE_WK in TSAM.Phase_4.

Access attempts not matching any of these rules will be rejected by SF.AC.

SF.LCM *Life cycle management*

SF.LCM manages life cycle state of the TOE. It does so by the following:

1. SF.LCM automatically initializes the life cycle state to TSAM.Phase_2 during applet installation in TSAM production.

2. When MK has been successfully initialized by R.Initializer in TSAM.Phase_2, SF.LCM will change the life cycle state to TSAM.Phase_3.
3. When TMD has been successfully written by R.Issuer in TSAM.Phase_3, SF.LCM will change the life cycle state to TSAM.Phase_4.

Life cycle state changes are irreversible. No other life cycle state changes are performed except the aforementioned ones.

SF.SDP *Stored data protection*

SF.SDP checks the integrity of RC, LCS and TMD stored in EEPROM. If an integrity violation is detected, the related command is cancelled and an output error code is provided to the external user.

1. Every time a value of RC, LCS or TMD is written to EEPROM, SF.SDP will generate a corresponding checksum in EEPROM.
2. On receipt of a command, SF.SDP will verify the checksum of LCS and check whether LCS has a valid value. If inconsistent checksum is detected or the value of LCS is out of range, SF.SDP will block processing of the command and return the corresponding error code.
3. If RC is accessed internally, SF.SDP will first of all verify the corresponding checksum. If inconsistent checksum is detected, SF.SDP blocks usage of RC and responds with a corresponding error code. This also indirectly blocks the usage of the corresponding MK.
4. If TMD is accessed internally, SF.SDP will first of all verify the corresponding checksum. If inconsistent checksum is detected, SF.SDP blocks usage of TMD and responds with a corresponding error code.

Furthermore SF.SDP stores MK and WKs in key objects of [JCOP41V231], and every time a value of MK or WK is written to EEPROM, the previous value is physically overwritten in the memory assigned to the corresponding key object.[2]

SF.USE_WK *Use of working keys*

SF.USE_WK provides the following cryptographic services applicable to TD (transaction data):

1. 3/DES encryption in ECB mode with key size of 112 bits according to ANSI X 9.52 TECB for encryption/decryption.
2. 3/DES decryption in ECB mode with key size of 112 bits according to ANSI X 9.52 TECB for encryption/decryption
3. 3/DES MAC generation in CBC mode with key size 112 bits according to ANSI X 9.9 with ANSI X 9.52 TCBC Encryption for MAC generation.

For each of the services there is one dedicated WK in the TOE. SF.USE_WK is only available in TSAM.Phase_4.

---

[2] Using key objects furthermore provides integrity protection for MK and WKs according to SF.Audit of [JCOP41V231ST], which locks the card session in case of corruption of check-summed objects.

SF.Embedded_Software (from [JCOP41V231ST])

The certified JavaCard platform (part of the TOE) features the following TSF. The exact formulation can be found in [JCOP41V231ST] (SF.Hardware from [JCOP41V231ST] is restated separately below):

1. Access control (SF.AccessControl)
2. Audit functionality (SF.Audit)
3. Cryptographic key management (SF.CryptoKey)
4. Cryptographic operation (SF.CryptoOperation), including random number generation according to [AIS 20] class K3 with SOF-high
5. Identification and authentication (SF.I&A)
6. Secure management of TOE resources (SF.SecureManagement)
7. PIN management (SF.PIN)
8. Transaction management (SF.Transaction)

SF.Hardware (from [JCOP41V231ST])

The certified hardware (part of the TOE) features the following TSF. The exact formulation can be found in [ST0348]:

1. Random Number Generator (F.RNG)
2. Triple-DES Co-processor (F.HW_DES)
3. AES Co-processor (F.HW_AES)
4. Control of Operating Conditions (F.OPC)
5. Protection against Physical Manipulation (F.PHY)
6. Logical Protection (F.LOG)
7. Protection of Mode Control (F.COMP)
8. Memory Access Control (F.MEM_ACC)
9. Special Function Register Access Control (F.SFR_ACC)

6.2 Strength of Function Claims

The minimum strength of function level claimed for this evaluation is SOF-high, therefore the following SOF-rateable security functions are also claimed to reach SOF-high. The security functions and corresponding permutational or probabilistic mechanisms to be SOF-rated are: **SF.AUT_GP** (Challenge-response authentication), **SF.CP_GP** (Cryptographic checksum verification), **SF.CP_MK** (Challenge-response authentication) and **SF.SDP** (Checksum verification). Any cryptographic algorithms in these functions will not be rated, but the rating will be performed with respect to protocols (e.g. whether challenge and response are sufficiently long).

The security functions **SF.AC** and **SF.LCM** are not based on any permutational or probabilistic mechanisms and, therefore, they don’t have to be rated. **SF.USE_WK** provides merely cryptographic mechanisms and, therefore, is also excluded from SOF rating.

Furthermore **SF.AUT_GP** and **SF.CP_MK** (both realizing challenge-response authentications) use random numbers. These random numbers are claimed to be conformant to [AIS20], class K3, SOF-high. (This conformance has already been evaluated for [JCOP41V231], and this composite TOE uses only random numbers from the corresponding evaluated random number generator).

6.3 Assurance Measures

The TOE is to fulfill the assurance requirements of assessment class ASE and of evaluation level EAL4 augmented by ADV_IMP.2 and AVA_VLA.4. The present document "Security Target" serves to fulfill the requirements according to ASE. Besides provision of the TOE (according to ATE_IND.2), the manufacturer will apply the following additional assurance measures within the frame of the evaluation, to evidently prove the fulfilling of the requirements according to EAL4 augmented by ADV_IMP.2 and AVA_VLA.4:

- Application of a compliant configuration management system and provision of corresponding documentation (according to ACM_AUT.1 and ACM_CAP.4)
- Application of secure delivery procedures and provision of delivery and operational documentation (according to ADO_DEL.2 and ADO_IGS.1)
- Provision of functional specification documentation (according to ADV_FSP.2)
- Provision of high-level design documentation (according to ADV_HLD.2)
- Provision of implementation representation (according to ADV_IMP.2)
- Provision of low-level design documentation (according to ADV_LLD.1)
- Provision of representation correspondence documentation (according to ADV_RCR.1)
- Provision of security policy modeling documentation (according to ADV_SPM.1)
- Provision of guidance documentation (according to AGD_ADM.1 and AGD_USR.1)
- Application of development security measures and provision of Life cycle support documentation (according to ALC_DVS.1, ALC_LCD.1, and ALC_TAT.1)
- Performance of functional tests and provision of corresponding test documentation (according to ATE_COV.2, ATE_DPT.1, and ATE_FUN.1)
- Provision of vulnerability assessment documentation (according to AVA_MSU.2, AVA_SOF.1, and AVA_VLA.4)

The assignment of the assurance measures to the assurance requirements (see section 5.3) is straight forward, as for all assurance components (with exception of the independent testing of the evaluator ATE_IND.2) corresponding documentation will be is provided.

7 PP claims

7.1 PP Reference

[JCOP41V231ST] and also this ST claim conformance to the following protection profile:

- Java Card System – Minimal Configuration Protection Profile, Version: 1.0b, August 2003 [JCSPP]

7.2 PP Additions and Refinements

See corresponding section of [JCOP41V231ST] and PP claims rationale in section 8.4 hereinafter.

8 Rationale

8.1 Security Objectives Rationale

	SO.REPLAY	SO.KEY_ACCESS	SO.TMD_ACCESS	SO.INTEGRITY & O.PROTECT_DATA	SO.TXN_SECURE	SO.SN	SOE.DLV	SOE.USE_DIAG	SOE.KEYS	SOE.DEV
T.KEY_ACCESS	X	X								
T.TMD_ACCESS	X		X							
T.INTEGRITY				X						
OSP.TXN_SECURE					X					
OSP.SN						X				
A.DLV							X			
A.USE_DIAG								X		
A.KEYS									X	
A.DEV										X

Table 8-1: Security objectives rationale

8.1.1 Traceability of the Security Objectives

SO.REPLAY directly traces back to the replay attack aspect of **T.TMD_ACCESS** and **T.KEY_ACCESS** concerning TMD writes and key initialization/updates/writes.

SO.TMD_ACCESS directly traces back to the authentication and integrity protection aspects of **T.TMD_ACCESS**.

SO.KEY_ACCESS directly traces back to the authentication, integrity protection and confidentiality protection aspects of **T.KEY_ACCESS**.

SO.TXN_SECURE directly traces back to **OSP.TXN_SECURE**, where introduction of R.POS_Terminal in **SO.TXN_SECURE** corresponds to no need for authentication as expressed in **OSP.TXN_SECURE**.

SO.SN directly traces back to **OSP.SN**, where introduction of R.POS_Terminal in **SO.SN** corresponds to no need for authentication as expressed in **OSP.SN**.

SO.INTEGRITY directly traces back to **T.INTEGRITY** concerning RC, LCS and TMD. Integrity protection for MK and WKs was already expressed by **O.PROTECT_DATA** of the platform, which traces back to **T.INTEGRITY** concerning MK and WKs.

SOE.DLV directly traces back to A.DLV (it is a re-statement of A.DLV).

SOE.USE_DIAG directly traces back to A.USE_DIAG (it is a re-statement of A.USE_DIAG).

SOE.KEYS directly traces back to A.KEYS (it is a re-statement of A.KEYS).

SOE.DEV directly traces back to A.DEV (it is a re-statement of A.DEV).

8.1.2 Coverage of the assumptions

A.DLV is covered by **SOE.DLV**, as **SOE.DLV** is a re-statement of **A.DLV**.

A.USE_DIAG is covered by **SOE.USE_DIAG**, as **SOE.USE_DIAG** is a re-statement of **A.USE_DIAG**.

A.DLV is covered by **SOE.DLV**, as **SOE.DLV** is a re-statement of **A.DLV**.

A.KEYS is covered by **SOE.KEYS**, as **SOE.KEYS** is a re-statement of **A.KEYS**.

8.1.3 Countering of the threats

T.INTEGRITY breaks down in two different aspects: (1) undetected integrity errors concerning MK, WKs, TMD, LCS and RC in storage, and (2) usage of corresponding corrupted data. Concerning TMD, LCS and RC both aspects are countered by **SO.INTEGRITY**, which defines as well integrity error detection concerning MK, WKs, TMD, LCS and RC in storage as the corresponding error response (prohibit use of corrupted data and give back error message). For MK and WKs, **T.INTEGRITY** is countered by **O.PROTECT_DATA** of the platform, which ensures integrity of any application keys (and other application data).

T.TMD_ACCESS is about (1) writes of TMD by roles other than R.Issuer, (2) undetected modification of TMD during authorized writes of TMD, and (3) replay of a TMD write. **SO.TMD_ACCESS** counters the first two of these aspects, as it defines (1) a secure mechanism for TMD writes that provides authentication of the R.Issuer and (2) integrity protection for the transferred TMD. The third aspect is countered by **SO.REPLAY**, which – among others – defines protection against replay attacks concerning TMD writes.

T.KEY_ACCESS is about (1) initialization and updates/writes of keys by roles other than R.Initializer and R.Issuer, respectively, (2) undetected modification of keys during transfer, (3) eavesdropping of keys during transfer, (4) reading out current keys from the TOE, (4) reading out previous key values from the TOE, and (5) replay of a key initialization/update/write. **SO.KEY_ACCESS** counters the first four of these aspects, as it defines (1) a secure mechanism for key initialization/updates/writes that provides authentication of the corresponding user, (2) integrity protection for the transferred keys, (3) confidentiality protection for the transferred keys, and (4) non-readability of keys. The fifth aspect is countered by **SO.REPLAY**, which – among others – defines protection against replay attacks concerning key initialization/updates/writes.

8.1.4 Coverage of the Organizational Security Policies

OSP.TXN_SECURE is covered by **SO.TXN_SECURE**, as **SO.TXN_SECURE** just defines the cryptographic functionalities as requested by **OSP.TXN.SECURE**. The aspect of **OSP.TXN_SECURE** that no authentication is needed is covered in **SO.TXN_SECURE** by the fact that the cryptographic functionalities shall be provided to R.POS_Terminal, which is an unauthenticated role.

OSP.SN is covered by **SO.SN**, as **SO.SN** just defines the serial number increment functionalities as requested by **OSP.SN**. The aspect of **OSP.SN** that no authentication is needed is covered in **SO.SN** by the fact that the increment functionalities shall be provided to R.POS_Terminal, which is an unauthenticated role.

8.1.5 Security Objectives Rationale from [JCOP41V231ST]

The following table is reproduced from [JCOP41V231ST] to illustrate the coverage of the threats by the security objectives concerning [JCOP41V231]. The corresponding justification text has not been reproduced here, please consult [JCOP41V231ST] if needed.

	O.PROTECT_DATA	O.OS_DECEIVE	O.SIDE_CHANNEL	O.FAULT_PROTECT	O.PHYSICAL	O.CARD-MANAGEMENT	O.SHRD_VAR_INTEG	O.SHRD_VAR_CONFID	O.FIREWALL	O.NATIVE	O.OPERATE	O.ALARM	O.RESOURCES	O.REALLOCATION	O.SID	O.SCP.IC	O.SCP.RECOVERY	O.SCP.SUPPORT	O.CIPHER	O.PIN-MNGT	O.KEY-MNGT	O.TRANSACTION	O.RND
T.ACCESS_DATA	X																						
T.OS_OPERATE	X										X												
T.OS_DECEIVE		X																					
T.LEAKAGE			X																				
T.FAULT				X																			
T.PHYSICAL					X											X							
T.CONFID-JCS-DATA T.INTEG-JCS-DATA						X			X		X	X			X		X	X					
T.CONFID-APPLI-DATA						X		X	X		X	X	X		X		X	X	X	X	X	X	
T.INTEG-APPLI-DATA						X	X		X		X	X	X		X		X	X	X	X	X	X	
T.SID.1						X			X						X								
T.SID.2									X		X				X		X	X					
T.NATIVE										X													
T.RESOURCES											X			X			X	X					
T.RND																							X

Table 8-2: Security objectives rationale from [JCOP41V231ST]

8.2 Security Requirements Rationale

	SO.REPLAY	SO.KEY_ACCESS	SO.TMD_ACCESS	SO.INTEGRITY	SO.TXN_SECURE	SOE.DLV	SOE.USE_DIAG	SOE.KEYS	SOE.DEV
FCS_CKM.4/TSAM		X							
FCS_COP.1/TSAM					X				
FDP_ACC.1/KEY		X							
FDP_ACC.1/TMD			X						
FDP_ACF.1/KEY		X							
FDP_ACF.1/TMD			X						
FDP_ITC.1/KEY		X							
FDP_ITC.1/TMD			X						
FDP_SDI.2/TSAM				X					
FDP_UCT.1/KEY		X							
FDP_UIT.1/TSAM	X	X	X						
FIA_AFL.1/TSAM		X	X						
FIA_UAU.1/TSAM		X	X						
FIA_UAU.4/TSAM		X	X						
FIA_UAU.5/TSAM		X	X						
FIA_UID.1/TSAM		X	X						
FMT_MSA.1/TSAM		X	X						
FMT_MSA.2/TSAM		X	X						
FMT_MSA.3/TSAM		X	X						
FMT_SMF.1/TSAM		X	X						
FMT_SMR.1/TSAM		X	X						
FTP_ITC.1/TSAM		X	X						
FCS_CKM.1/ENV								X	
FDP_UCT.1/ENV							X		
FDP_UIT.1/ENV							X		
FTP_ITC.1/ENV							X		
FDP_IFC.2/BCV									X
FDP_IFF.2/BCV									X
FMT_MSA.1/BCV									X
FMT_MSA.2/BCV									X
FMT_MSA.3/BCV									X
FMT_SMF.1/BCV									X
FMT_SMR.1/BCV									X
FRU_RSA.1/BCV									X

Table 8-3: Security requirements rationale

8.2.1 Fulfilment of security objectives

SO.REPLAY is met by FDP_UIT.1/TSAM, as this requires replay protection concerning key initialization/updates/writes as defined in SO.REPLAY.

SO.KEY_ACCESS defines (1) access control for key initialization, writes and updates, and (2) confidentiality protection and (3) integrity protection during those operations.

The first aspect is met by the access control requirements FDP_ACC.1/KEY and FDP_ACF.1/KEY. Management for the governing security attribute life-cycle state is provided by FMT_MSA.1/TSAM, FMT_MSA.2/TSAM, FMT_MSA.3/TSAM and FMT_SMF.1/TSAM. The identification and authentication and ability to distinguish roles, which are a precondition for performing the access control, are provided by FIA_AFL.1/TSAM, FIA_UAU.1/TSAM, FIA_UAU.4/TSAM, FIA_UAU.5/TSAM and FIA_UID.1/TSAM and FMT_SMR.1/TSAM.

The second aspect, key confidentiality, is met by FDP_UCT.1/KEY and FTP_ITC.1/TSAM, furthermore FCS_CKM.4/TSAM supports it by ensuring that values of previous keys are no longer physically available after a key update/write.

The third aspect, key integrity, is provided by FDP_UIT.1/TSAM and FTP_ITC.1/TSAM.

Finally, the key import operations are defined by FDP_ITC.1/KEY.

SO.TMD_ACCESS defines (1) access control for TMD writes, and (2) integrity protection during those operations.

The first aspect is met by the access control requirements FDP_ACC.1/TMD and FDP_ACF.1/TMD. Management for the governing security attribute life-cycle state is provided by FMT_MSA.1/TSAM, FMT_MSA.2/TSAM, FMT_MSA.3/TSAM and FMT_SMF.1/TSAM. The identification and authentication and ability to distinguish roles, which are a precondition for performing the access control, are provided by FIA_AFL.1/TSAM, FIA_UAU.1/TSAM, FIA_UAU.5/TSAM, FIA_UID.1/TSAM and FMT_SMF.1/TSAM.

The second aspect, TMD integrity, is provided by FDP_UIT.1/TSAM and FTP_ITC.1/TSAM.

Finally, the TMD import operation is defined by FDP_ITC.1/TMD.

SO.INTEGRITY defines (1) integrity detection concerning RC, LCS and TMD in storage and (2) prevention to use corrupted data and provision of an error message.

The first aspect is met by FDP_SDI.2.1/TSAM, which requests the corresponding stored data integrity monitoring. The second aspect is met by FDP_SDI.2.2/TSAM, which requests a message to the user and dedicated response actions that prevent usage of the corrupted data.

SO.TXN_SECURE defines cryptographic services of encryption, decryption and MAC generation using 3/DES. This is directly met by FCS_COP.1/TSAM.

SOE.DLV is purely related to non-IT environmental aspects (organizational measures concerning verification of delivered items), therefore there are no related SFRs.

SOE.USE_DIAG defines the capabilities of the IT environment concerning integrity and confidentiality protection during data transfer. This is directly met by FDP_UCT.1/ENV, FDP_UIT.1/ENV and FTP_ITC.1/ENV.

SOE.KEYS defines random number generation as the method to generate keys. This is directly met by FCS_CKM.1/ENV. The remaining aspects of

SOE.KEYS (confidentiality and integrity protection of keys in the environment) may be solely related to non-IT measures, therefore there are no related SFRs.

SOE.DEV defines performance of byte code verification during development. This is met by the FDP_IFC.2/BCV, FDP_IFF.2/BCV, FMT_MSA.1/BCV, FMT_MSA.2/BCV, FMT_MSA.3/BCV, FMT_SMF.1/BCV, FMT_SMR.1/BCV, FRU_RSA.1/BCV concerning byte code verification. The remaining aspects of SOE.DEV (no native code loading, no delivery of GlobalPlatform keys and general development security aspects) may be solely related to non-IT measures, therefore there are no related SFRs.

8.2.2 Traceability of the Security Functional Requirements

FCS_CKM.4/TSAM requires physical overwriting of previous key values keys during updates/writes and therefore traces back to the aspect of SO.KEY_ACCESS that previous key values shall not be accessible.

FCS_COP.1/TSAM requires the cryptographic functions needed to secure transactions and therefore traces back to SO.TXN_SECURE.

FDP_ACC.1/KEY and FDP_ACF.1/KEY define access control concerning keys and therefore trace back to SO.KEY_ACCESS.

FDP_ACC.1/TMD and FDP_ACF.1/TMD define access control concerning TMD and therefore trace back to SO.TMD_ACCESS.

FDP_ITC.1/KEY defines details about the key import operation and therefore traces back to SO.KEY_ACCESS.

FDP_ITC.1/TMD defines details about the TMD import operation and therefore traces back to SO.TMD_ACCESS.

FDP_SDI.2/TSAM requires integrity protection concerning stored RC, LCS and TMD, and therefore traces back to SO.INTEGRITY.

FDP_UCT.1/KEY requires confidentiality protection concerning transfer of keys and therefore traces back to SO.KEY_ACCESS.

FDP_UIT.1/TSAM requires integrity and replay protection concerning transfer of keys and TMD and therefore traces back to SO.KEY_ACCESS, SO.TMD_ACCESS and SO.REPLAY.

FIA_AFL.1/TSAM, FIA_UAU.1/TSAM, FIA_UAU.4/TSAM, FIA_UAU.5/TSAM and FIA_UID.1/TSAM define requirements about identification and authentication necessary as a precondition for the access control about keys and TMD, and therefore trace back to SO.KEY_ACCESS and SO.TMD_ACCESS.

FMT_MSA.1/TSAM, FMT_MSA.2/TSAM, FMT_MSA.3/TSAM and FMT_SMF.1/TSAM define requirements about the management of the life-cycle state, which is the governing security attribute for access control for keys and TMD, and therefore traces back to SO.KEY_ACCESS and SO.TMD_ACCESS.

FMT_SMR.1/TSAM requires the ability to distinguish roles, which is a precondition for access control for keys and TMD, and therefore traces back to SO.KEY_ACCESS and SO.TMD_ACCESS

FTP_ITC.1/TSAM defines a trusted channel that provides confidentiality and/or integrity protection for key/TMD transfer, and therefore traces back to SO.KEY_ACCESS and SO.TMD_ACCESS.

FCS_CKM.1/ENV defines random number generation as key generation algorithm to be used, and therefore traces back to the corresponding aspect of SOE.KEYS.

FDP_UCT.1/ENV, FDP_UIT.1/ENV and FTP_ITC.1/ENV define requirements for remote IT products in the environment concerning confidentiality and integrity protection during data transfer to the TOE, and therefore trace back to SOE.USE_DIAG.

FDP_IFC.2/BCV, FDP_IFF.2/BCV, FMT_MSA.1/BCV, FMT_MSA.2/BCV, FMT_MSA.3/BCV, FMT_SMF.1/BCV, FMT_SMR.1/BCV, FRU_RSA.1/BCV define requirements concerning byte code verification, and therefore trace back to the corresponding aspect of SOE.DEV.

8.2.3 Suitability of Security Assurance Requirements

As the TOE shall be used in a financial context and its assets will have high financial value, a corresponding high level of robustness of and confidence in the TOE is required. Therefore as assurance requirements EAL4 augmented by ADV_IMP.2 and AVA_VLA.4 have been chosen.

Confidence will be provided, as EAL4 requires a thorough evaluation, in particular of the design of the TOE (which even has been extended by the augmentation of ADV_IMP.2).

Sufficient robustness of the TOE against penetration attacks shall be provided by application of AVA_VLA.4, which provides for a systematic vulnerability analysis and finally for a TOE being resistant even to attackers owing a high attack potential.

8.2.4 Fulfillment of dependencies

SFR used	Dependencies acc. to CC	Fulfilled by
FCS_CKM.4/TSAM	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	FDP_ITC.1/KEY FMT_MSA.2/TSAM
FCS_COP.1/TSAM	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FDP_ITC.1/KEY FCS_CKM.4/TSAM FMT_MSA.2/TSAM
FDP_ACC.1/KEY	FDP_ACF.1	FDP_ACF.1/KEY
FDP_ACC.1/TMD	FDP_ACF.1	FDP_ACF.1/TMD
FDP_ACF.1/KEY	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KEY FMT_MSA.3/TSAM
FDP_ACF.1/TMD	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/TMD FMT_MSA.3/TSAM
FDP_ITC.1/KEY	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	FDP_ACC.1/KEY FMT_MSA.3/TSAM
FDP_ITC.1/TMD	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	FDP_ACC.1/TMD FMT_MSA.3/TSAM

SFR used	Dependencies acc. to CC	Fulfilled by
FDP_SDI.2/TSAM	No dependencies	Not applicable
FDP_UCT.1/KEY	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1]	FTP_ITC.1/TSAM FDP_ACC.1/KEY
FDP_UIT.1/TSAM	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	FDP_ACC.1/KEY, FDP_ACC.1/TMD FTP_ITC.1/TSAM
FIA_AFL.1/TSAM	FIA_UAU.1	FIA_UAU.1/TSAM
FIA_UAU.1/TSAM	FIA_UID.1	FIA_UID.1/TSAM
FIA_UAU.4/TSAM	No dependencies	Not applicable
FIA_UAU.5/TSAM	No dependencies	Not applicable
FIA_UID.1/TSAM	No dependencies	Not applicable
FMT_MSA.1/TSAM	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KEY, FDP_ACC.1/TMD FMT_SMR.1/TSAM FMT_SMF.1/TSAM
FMT_MSA.2/TSAM	ADV_SPM.1 [FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	ADV_SPM.1 FDP_ACC.1/KEY, FDP_ACC.1/TMD FMT_SMR.1/TSAM FMT_SMF.1/TSAM
FMT_MSA.3/TSAM	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/TSAM FMT_SMR.1/TSAM
FMT_SMF.1/TSAM	No dependencies	Not applicable
FMT_SMR.1/TSAM	FIA_UID.1	FIA_UID.1/TSAM
FTP_ITC.1/TSAM	No dependencies	Not applicable

Table 8-4: Fulfillment of TOE SFR dependencies

Concerning the security assurance requirements all dependencies are fulfilled, as

- all dependencies within an evaluation assurance level (here: EAL4) are automatically fulfilled,
- the dependencies of the augmented component ADV_IMP.2 (i.e. ADV_LLD.1, ADV_RCR.1 and ALC_TAT.1) are already satisfied within EAL4,
- and the dependencies of the augmented component AVA_VLA.4 (i.e. ADV_FSP.1, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1 and AGD_USR.1) are already satisfied within EAL4.

8.2.5 Suitability of minimum strength of function (SoF) level

As the TOE shall be used in a financial context and its assets will have high financial value, the TOE also shall be highly resistant against attacks on its functions. The protection against attacks with a high attack potential dictates a strength of function rating of “high”.

8.3 TOE Summary Specification Rationale

	SF.AUT_GP	SF.CP_GP	SF.CP_MK	SF.AC	SF.LCM	SF.SDP	SF.USE_WK
FCS_CKM.4/TSAM						X	
FCS_COP.1/TSAM							X
FDP_ACC.1/KEY				X			
FDP_ACC.1/TMD				X			
FDP_ACF.1/KEY				X			
FDP_ACF.1/TMD				X			
FDP_ITC.1/KEY				X	X		
FDP_ITC.1/TMD				X	X		
FDP_SDI.2/TSAM						X	
FDP_UCT.1/KEY		X	X				
FDP_UIT.1/TSAM		X	X				
FIA_AFL.1/TSAM			X				
FIA_UAU.1/TSAM				X			
FIA_UAU.4/TSAM	X						
FIA_UAU.5/TSAM	X	X	X				
FIA_UID.1/TSAM				X			
FMT_MSA.1/TSAM					X		
FMT_MSA.2/TSAM					X		
FMT_MSA.3/TSAM					X		
FMT_SMF.1/TSAM					X		
FMT_SMR.1/TSAM				X			
FTP_ITC.1/TSAM	X	X					

Table 8-5: TSS rationale

8.3.1 Traceability and Satisfaction of the TOE SFRs

The following table shows that the TOE security functions satisfy the corresponding SFRs, that all SFRs are addressed and that there is no aspect of a security function that cannot be traced back to an SFR. This is achieved by breaking down the security functions in individual statements and mapping each statement to the corresponding SFR(s).

Statements of Security Function	Fulfilled SFR(s)
SF.AUT_GP: SF.AUT_GP will authenticate the user by a challenge-response mechanism using GlobalPlatform keys. …	**FIA_UAU.5.1/TSAM:** The TSF shall provide [GlobalPlatform card manager authentication, …] to support user authentication.
SF.AUT_GP: … For each authentication attempt, SF.AUT_GP will present a new random number as a challenge. …	**FIA_UAU.4.1/TSAM:** The TSF shall prevent reuse of authentication data related to [*GlobalPlatform card manager authentication, …*].
SF.AUT_GP: … Only if the user provides the corresponding correct response, the user is authenticated as the initializer (R.Initializer). …	**FIA_UAU.5.2/TSAM:** The TSF shall authenticate any user's claimed identity according to the [following rules: … GlobalPlatform card manager authentication is used for authentication of

Statements of Security Function	Fulfilled SFR(s)
	R.Initializer …]
SF.AUT_GP: … In case of a successful authentication, SF.AUT_GP will establish session keys that are later on used by SF.CP_GP. …	**FTP_ITC.1.1/TSAM** 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/TSAM** The TSF shall permit [the remote trusted IT product] to initiate communication via the trusted channel. **FTP_ITC.1.3/TSAM** The TSF shall initiate communication via the trusted channel for [performing initialize, … of MK …].
SF.AUT_GP: … SF.AUT_GP is only available in TSAM.Phase_2.	**FIA_UAU.5.2/TSAM:** The TSF shall authenticate any user's claimed identity according to the [following rules: … GlobalPlatform card manager authentication is used for authentication … in TSAM.Phase_2. …]
SF.CP_GP: SF.CP_GP provides confidentiality … protection of communication data between the user and the TOE. This is done by decryption … using session keys. The corresponding session keys are established after a successful authentication by SF.AUT_GP. …	**FDP_UCT.1.1/KEY** The TSF shall enforce the [Key Access SFP] to be able to [receive] objects in a manner protected from unauthorised disclosure. **FTP_ITC.1.1/TSAM** 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 … disclosure. **FTP_ITC.1.2/TSAM** The TSF shall permit [the remote trusted IT product] to initiate communication via the trusted channel. **FTP_ITC.1.3/TSAM** The TSF shall initiate communication via the trusted channel for [performing initialize, … of MK …].
SF.CP_GP: SF.CP_GP provides … integrity protection of communication data between the user and the TOE. This is done by … verification of cryptographic checksum using session keys. The corresponding session keys are established after a successful authentication by SF.AUT_GP. …	**FDP_UIT.1.1/TSAM** The TSF shall enforce the [Key Access SFP and TMD Access SFP] to be able to [receive] user data in a manner protected from [modification, insertion, replay] errors. **FDP_UIT.1.2/TSAM** The TSF shall be able to determine on receipt of user data, whether [modification, insertion, replay] has occurred. **FTP_ITC.1.1/TSAM** 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 ... **FTP_ITC.1.2/TSAM** The TSF shall permit [the remote trusted IT product] to initiate communication via the trusted channel.

Statements of Security Function	Fulfilled SFR(s)
	FTP_ITC.1.3/TSAM The TSF shall initiate communication via the trusted channel for [performing initialize, … of MK …].
SF.CP_GP: … SF.CP_GP is only available in TSAM.Phase_2.	**FIA_UAU.5.2/TSAM** The TSF shall authenticate any user's claimed identity according to the [following rules: 1. GlobalPlatform card manager authentication is used for authentication of R.Initializer in TSAM.Phase_2. …]. (Remark: SF.CP_GP uses the same GlobalPlatform functionality as SF.AUT_GP.)
SF.CP_MK: SF.CP_MK assures integrity, authenticity … confidentiality of communication data between the user and the TOE for a single command. This is done by MAC verification … using session keys which are only valid for this command. To do so, SF.CP_MK performs the following … steps: 1. For establishing the session keys, a random number RN is provided by SF.CP_MK to the user as the very first step. 2. SF.CP_MK receives the command from the user. … 4. SF.CP_MK generates the session key for MAC verification by encrypting RN with MK. SF.CP_MK verifies the MAC within the command. If verification fails, it … returns an error code and stops processing. Otherwise, the issuer (R.Issuer) is authenticated, and SF.CP_MK … continues with the next step. …	**FDP_UIT.1.1/TSAM** The TSF shall enforce the [Key Access SFP and TMD Access SFP] to be able to [receive] user data in a manner protected from [modification, insertion, replay] errors. **FDP_UIT.1.2/TSAM** The TSF shall be able to determine on receipt of user data, whether [modification, insertion, replay] has occurred.
SF.CP_MK: SF.CP_MK assures … optionally confidentiality of communication data between the user and the TOE for a single command. This is done by … decryption using session keys which are only valid for this command. To do so, SF.CP_MK performs the following … steps: 1. For establishing the session keys, a random number RN is provided by SF.CP_MK to the user as the very first step. 2. SF.CP_MK receives the command from the user. ... 5. If the command includes encrypted data, SF.CP_MK generates the session key for decryption by encrypting the inverse of RN with MK and SF.CP_MK decrypts the encrypted data. …	**FDP_UCT.1.1/KEY** The TSF shall enforce the [Key Access SFP] to be able to [receive] objects in a manner protected from unauthorised disclosure.
SF.CP_MK: … 3. SF.CP_MK checks the value of RC. If it is equal to 3, SF.CP_MK returns an error code and stops processing. Otherwise, it continues with the next step. 4. … If verification fails, it increases RC, returns an error code and stops processing. Otherwise, … SF.CP_MK resets RC to zero and	**FIA_AFL.1.1/TSAM** The TSF shall detect when [three consecutive] unsuccessful authentication attempts occur related to [authentication with MK]. **FIA_AFL.1.2/TSAM** When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [no longer

Statements of Security Function	Fulfilled SFR(s)
continues with the next step. …	allow authentication with MK].
SF.CP_MK: …SF.CP_MK is only available in TSAM.Phase_3 and TSAM.Phase_4.	**FIA_UAU.5.2/TSAM** The TSF shall authenticate any user's claimed identity according to the [following rules: … 2. Authentication with MK is used for authentication of R.Issuer in TSAM.Phase_3 and TSAM.Phase_4].
SF.AC: SF.AC enforces access control rules based on commands, user roles and life cycle state. …	**FDP_ACC.1.1/KEY** The TSF shall enforce the [Key Access SFP] on [subjects: users, objects: MK, WKs and operation: initialize, first update, update, write, read and use]. **FDP_ACC.1.1/TMD** The TSF shall enforce the [TMD Access SFP] on [subjects: users, objects: TMD and operation: read, write and increment].
SF.AC: … For commands needing authentication, SF.AC identifies user roles R.Initializer and R.Issuer with SF.AUT_GP and SF.CP_MK, respectively. For commands not needing authentication, SF.AC identifies the user role as R.POS_Terminal. …	**FMT_SMR.1.1/TSAM** The TSF shall maintain the roles [R.Initializer, R.Issuer and R.POS_Terminal]. **FMT_SMR.1.2/TSAM** The TSF shall be able to associate users with roles. **FIA_UAU.1.1/TSAM** The TSF shall allow [encryption, decryption and MAC generation by corresponding WK, reading TMD, incrementing TSN and/or BSN of TMD] on behalf of the user to be performed before the user is authenticated. **FIA_UAU.1.2/TSAM** The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. **FIA_UID.1.1/TSAM** The TSF shall allow [encryption, decryption and MAC generation by corresponding WK, reading TMD, incrementing TSN and/or BSN of TMD] on behalf of the user to be performed before the user is identified. **FIA_UID.1.2/TSAM** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. (see also SF.AUT_GP and SF.CP_MK)
SF.AC: … The following is SF.AC-enforced access control rules: 1. The initializer (R.Initializer) is allowed to initialize MK in TSAM.Phase_2. 2. The issuer (R.Issuer) is allowed to perform first update of MK in TSAM.Phase_3. The issuer is also allowed to perform updates of MK and writes of WKs in TSAM.Phase_4. 3. No user can read any of the MK and WKs out of the TOE. 4. The issuer (R.Issuer) is allowed to write TMD in TSAM.Phase_3. 5. The user R.POS_Terminal is allowed to	**FDP_ACF.1.1/KEY** The TSF shall enforce the [Key Access SFP] to objects based on the following: [subject attribute: user role {R.Initializer, R.Issuer, R.POS_Terminal} and object attribute: life cycle state { TSAM.Phase_2, TSAM.Phase_3, TSAM.Phase_4}]. **FDP_ACF.1.2/KEY** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [1. A user with user role {R.Initializer} is allowed to initialize the MK if the life cycle state is {TSAM.Phase_2}. 2. A user with user role {R.Issuer} is

Statements of Security Function	Fulfilled SFR(s)
read TMD out of the TOE in TSAM.Phase_4. 6. The user R.POS_Terminal is allowed to increment TSN of TMD in TSAM.Phase_4 unless the value of TSN is equal to 999999. 7. The user R.POS_Terminal is allowed to increment BSN of TMD in TSAM.Phase_4 unless the value of BSN is equal to 9999. 8. The user R.POS_Terminal is allowed to use WKs according to SF.USE_WK in TSAM.Phase_4. Access attempts not matching any of these rules will be rejected by SF.AC.	allowed to do first update of the MK if the life cycle state is {TSAM.Phase_3}. 3. A user with user role {R.Issuer} is allowed to do updates of the MK if the life cycle state is {TSAM.Phase_4}. 4. A user with user role {R.Issuer} is allowed to do writes of the WK if the life cycle state is {TSAM.Phase_4}. 5. A user with user role {R.POS_Terminal} is allowed to use the WK if the life cycle state is {TSAM.Phase_4}.] **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 [rule that no user can read any of the MK and WKs out of the TOE]. **FDP_ACF.1.1/TMD** The TSF shall enforce the [TMD Access SFP] to objects based on the following: [subject attribute: user role {R.Issuer, R.POS_Terminal} and object attribute: life cycle state {TSAM.Phase_3, TSAM.Phase_4}]. **FDP_ACF.1.2/TMD** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [1. A user with user role {R.Issuer} is allowed to write TMD if the life cycle state is {TSAM.Phase_3}. 2. A user with user role {R.POS_Terminal} is allowed to read TMD and increment TSN/BSN of TMD if the life cycle state is {TSAM.Phase_4}.] **FDP_ACF.1.3/TMD** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [no other rule]. **FDP_ACF.1.4/TMD** The TSF shall explicitly deny access of subjects to objects based on the [following rules: 1. Increment of TSN is denied if the value of TSN is equal to 999999. 2. Increment of BSN is denied if the value of BSN is equal to 9999.].
SF.AC: … 1. The initializer (R.Initializer) is allowed to initialize MK ... 2. The issuer (R.Issuer) is allowed to perform first update of MK ... The issuer is also allowed to perform updates of MK and writes of WKs …	**FDP_ITC.1.1/KEY** The TSF shall enforce the [Key Access SFP] when importing user data, controlled under the SFP, from outside of the TSC. **FDP_ITC.1.2/KEY** The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.
SF.AC: … 4. The issuer (R.Issuer) is allowed	**FDP_ITC.1.1/TMD** The TSF shall enforce the

Statements of Security Function	Fulfilled SFR(s)
to write TMD …	[TMD Access SFP] when importing user data, controlled under the SFP, from outside of the TSC. **FDP_ITC.1.2/TMD** The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.
SF.LCM: SF.LCM provides management of the life cycle state of the TOE, … Life cycle state changes are irreversible. No other life cycle state changes are performed except the aforementioned ones.	**FMT_SMF.1.1/TSAM** The TSF shall be capable of performing the following security management functions: [modification of the life state according to FMT_MSA.1.1/TSAM, FDP_ITC.1.3 /KEY and FDP_ITC.1.3 /TMD].
SF.LCM: … It does so by the following: 1. SF.LCM automatically initializes the life cycle state to TSAM.Phase_2 during applet installation in TSAM production. …	**FMT_MSA.3.1/TSAM** The TSF shall enforce the [Key Access SFP and TMD Access SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP. **FMT_MSA.3.2/TSAM** The TSF shall allow the [nobody] to specify alternative initial values to override the default values when an object or information is created. **FMT_MSA.2.1/TSAM** The TSF shall ensure that only secure values are accepted for security attributes.
SF.LCM: … 2. When MK has been successfully initialized by R.Initializer in TSAM.Phase_2, SF.LCM will change the life cycle state to TSAM.Phase_3. …	**FDP_ITC.1.3/KEY** The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [1. After import of MK by initialize operation, the security attribute life cycle state shall change from TSAM.Phase_2 to TSAM.Phase_3.] **FMT_MSA.1.1/TSAM** The TSF shall enforce the [Key Access SFP and TMD Access SFP] to restrict the ability to [modify] the security attributes [life cycle state] to [R.Initializer and …]. (Remark: SF.AC enforces that initialization can only be done by R.Initializer, and SF.LCM links the LCS transition to initialization operation.) **FMT_MSA.2.1/TSAM** The TSF shall ensure that only secure values are accepted for security attributes.
SF.LCM: … 3. When TMD has been successfully written by R.Issuer in TSAM.Phase_3, SF.LCM will change the life cycle state to TSAM.Phase_4. …	**FDP_ITC.1.3/TMD** The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [1. After import of TMD by write operation, the security attribute life cycle state shall change from TSAM.Phase_3 to TSAM.Phase_4.] **FMT_MSA.1.1/TSAM** The TSF shall enforce the [Key Access SFP and TMD Access SFP] to restrict the ability to [modify] the security attributes [life cycle state] to [R.Initializer and R.Issuer]. (Remark: SF.AC enforces that TMD writing can only be done by R.Issuer, and SF.LCM links the LCS transition to the TMD writing operation.) **FMT_MSA.2.1/TSAM** The TSF shall ensure that

Statements of Security Function	Fulfilled SFR(s)
	only secure values are accepted for security attributes.
SF.SDP: SF.SDP checks the integrity of RC, LCS and TMD stored in EEPROM. If an integrity violation is detected, the related command is cancelled and an output error code is provided to the external user. 1. Every time a value of RC, LCS or TMD is written to EEPROM, SF.SDP will generate a corresponding checksum in EEPROM. 2. On receipt of a command, SF.SDP will verify the checksum of LCS and check whether LCS has a valid value. If inconsistent checksum is detected or the value of LCS is out of range, SF.SDP will block processing of the command and return the corresponding error code. 3. If RC is accessed internally, SF.SDP will first of all verify the corresponding checksum. If inconsistent checksum is detected, SF.SDP blocks usage of RC and responds with a corresponding error code. This also indirectly blocks the usage of the corresponding MK. 4. If TMD is accessed internally, SF.SDP will first of all verify the corresponding checksum. If inconsistent checksum is detected, SF.SDP blocks usage of TMD and responds with a corresponding error code. …	**FDP_SDI.2.1/TSAM** The TSF shall monitor user data stored within the TSC for [integrity errors] on all objects, based on the following attributes [checksum for TMD, LCS and RC]. **FDP_SDI.2.2/TSAM** Upon detection of a data integrity error, the TSF shall [inform the user and perform the actions in Table 5 1 depending on which object is incurred in the data integrity error].
SF.SDP: … Furthermore SF.SDP stores MK and WKs in key objects of [JCOP41V231], and every time a value of MK or WK is written to EEPROM, the previous value is physically overwritten in the memory assigned to the corresponding key object.	**FCS_CKM.4.1/TSAM** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [previous MK and WKs are physically overwritten by new keys] that meets the following: [none].
SF.USE_WK: SF.USE_WK provides the following cryptographic services applicable to TD (transaction data): 1. 3/DES encryption in ECB mode with key size of 112 bits according to ANSI X 9.52 TECB for encryption/decryption. 2. 3/DES decryption in ECB mode with key size of 112 bits according to ANSI X 9.52 TECB for encryption/decryption 3. 3/DES MAC generation in CBC mode with key size 112 bits according to ANSI X 9.9 with ANSI X 9.52 TCBC Encryption for MAC generation. For each of the services there is one dedicated WK in the TOE. SF.USE_WK is only available in TSAM.Phase_4.	**FCS_COP.1.1/TSAM** The TSF shall perform [encryption, decryption, MAC generation for TD with dedicated keys in TSAM.Phase.4] in accordance with a specified cryptographic algorithm [3/DES in ECB or CBC mode] and cryptographic key sizes [112 bits] that meet the following: [ANSI X 9.52 TECB for encryption/decryption, ANSI X 9.9 with ANSI X 9.52 TCBC Encryption for MAC generation].

1469 **Table 8-6: Traceability and Satisfaction of the TOE SFRs**

8.3.2 Mutual Support of the Security Functions

Based on the results of the security requirements rationale, which has shown that the set of TOE SFRs forms a mutually supportive whole to fulfil the TOE security objectives, according to [CEM] § 491 the remaining question here is whether the additional information included in the security functions introduces no potential security weakness, such as possibilities to bypass tamper with, or deactivate other IT security functions. Based on the mapping of security functions and TOE SFRs in section 8.3.1 above, in the following this additional information in the security functions is listed, and its impact on the mutual support is discussed:

Security Function	**Additional Information in SF compared to corresponding SFR(s)**	**Impact on Mutual Support of Security Functions**
SF.AUT_GP	Mechanism *challenge-response authentication* (with a new random number for each authentication attempt) introduced.	Only impact on SF_AUT_GP itself; not in conflict with any other SF.
	Mechanism *session key generation* introduced; link to SF.CP_GP introduced.	Impact on SF.AUT_GP itself; is a precondition to support SF_CP_GP; not in conflict with any other SF.
SF.CP_GP	Mechanism *decryption using session keys* introduced; link to SF.AUT_GP introduced.	Impact on SF_CP_GP itself; is supported by session key generation of SF.AUT_GP; not in conflict with any other SF.
	Mechanism *verification of cryptographic checksum using session keys* introduced; link to SF.AUT_GP introduced.	Impact on SF_CP_GP itself; is supported by session key generation of SF.AUT_GP; not in conflict with any other SF.
SF.CP_MK	Mechanism *command-wise MAC verification using session keys* introduced.	Only impact on SF_CP_MK itself; not in conflict with any other SF.
	Mechanism *command- wise decryption using session keys* introduced.	Only impact on SF_CP_MK itself; not in conflict with any other SF.
SF.AC	None.	N/A
SF.LCM	None.	N/A
SF.SDP	Details concerning kind and time of checksum verification introduced.	Impact on SF.SDP and all other SFs, as the TOE will stop a running session in case of a detected integrity error, but not in conflict with the other SFs or the corresponding security objectives.
SF.USE_WK	None.	N/A.

Table 8-7: Analysis of Mutual Support of the SFs

8.3.3 Validity of SOF-claims

In section 5 the minimum strength of function level for this TOE is claimed to be SOF-high. This is consistent to the strength of function claims in section 6.2, which is also SOF-high for all the rateable functions.

8.3.4 Compliance of assurance measures

The assurance measures as stated in section 6.3 address all aspects of the assurance requirements of the chosen set EAL4+ and are therefore compliant in principle. Whether this is actually the case will be inspected during evaluation of the corresponding evidence.

8.4 PP Claims Rationale

This security target claims conformance to the protection profile [JCSPP], “Minimal Configuration”, as the security target of the underlying platform, [JCOP41V231ST], does. The following subsections show that this ST is in fact comformant to [JCSPP], “Minimal Configuration”

8.4.1 PP Conformance concerning Assumptions

All assumptions from [JCSPP] “Minimal Configuration” (which are restated in [JCOP41V231ST]) are covered by this ST the following way:

<table>
<tr><th>Assumption from [JCSPP] (and [JCOP41V231ST])</th><th>Coverage in this ST</th></tr>
<tr><td>A.NATIVE (native code APIs/applications ensure that security policies/objectives are not violated)</td><td>by A.DEV (during TSAM production/operation no native code will be loaded into the smart card controller)</td></tr>
<tr><td>A.NO-DELETION (no deletion of installed applets/packages possible)</td><td rowspan="2">by A.DEV (GlobalPlatform keys are not delivered, therefore no applet management possible)</td></tr>
<tr><td>A.NO-INSTALL (no post-issuance installation of applets)</td></tr>
<tr><td>A.VERIFICATION (byte code is verified to ensure its validity at execution time)</td><td>by A.DEV (byte code verification will be performed during TSAM development/production)</td></tr>
</table>

Table 8-8: Coverage of Assumptions from [JCSPP] “Minimal Configuration”

8.4.2 PP Conformance concerning Threats

All threats from [JCSPP] “Minimal Configuration” (regarding the Java Card platform level) are contained in [JCOP41V231ST] and also this ST, see Table 3-1 hereinbefore. The additional threats in this ST regard the TSAM application level and are not in contradiction to the ones from [JCSPP].

8.4.3 PP Conformance concerning Organizational Security Policies

There is no OSP in [JCSPP] “Minimal Configuration”.

8.4.4 PP Conformance concerning Security Objectives for the TOE

The security objectives for the TOE from [JCSPP] Minimal Configuration, all regarding the Java Card platform level, are contained in [JCOP41V231ST] and also this ST, see section 4.1.2 hereinbefore. The additional security objectives for the TOE in this ST regard the TSAM application level and are not in contradiction to the ones from [JCSPP].

8.4.5 PP Conformance concerning Security Objectives for the Environment

All security objectives for the environment from [JCSPP] “Minimal Configuration” (which are restated in [JCOP41V231ST]) are covered by this ST the following way:

Security Objective from [JCSPP] (and [JCOP41V231ST])	**Coverage in this ST**
OE.NATIVE (native code APIs/applications ensure that security policies/objectives are not violated)	by SOE.DEV (during TSAM production/operation no native code will be loaded into the smart card controller)
OE.NO-DELETION (no deletion of installed applets/packages possible)	by SOE.DEV (GlobalPlatform keys are not delivered, therefore no applet management possible)
OE.NO-INSTALL (no post-issuance installation of applets)	
OE.VERIFICATION (byte code is verified to ensure its validity at execution time)	by SOE.DEV (byte code verification will be performed during TSAM development/production)

Table 8-9: Coverage of Environment Security Objectives from [JCSPP]

8.4.6 PP Conformance concerning SFRs for the TOE

The SFRs for the TOE from [JCSPP] Minimal Configuration, all regarding the Java Card platform level, are contained in [JCOP41V231ST] and also this ST, see section 5.2 hereinbefore. The additional SFRs for the TOE in this ST regard the TSAM application level and are not in contradiction to the ones from [JCSPP].

Furthermore [JCSPP] defines the minimum strength of function level to be SoF-medium. Here this claim is exceeded by using SoF-high, therefore this ST is conformant to [JCSPP] concerning the minimum SoF-claim.

8.4.7 PP Conformance concerning SARs

[JCSPP] claims conformance to EAL4 augmented by AVA_VLA.3 and ADV_IMP.2. In this ST conformance to EAL4 augmented by AVA_VLA.4 and ADV_IMP.2 is claimed. As AVA_VLA.4 is hierarchical to AVA_VLA.3, this ST is conformant to [JCSPP] concerning the security assurance requirements.

8.4.8 PP Conformance concerning SFRs for the IT Environment

Environment SFRs from [JCSPP]	**Coverage in this ST**
[JCSPP] section 5.1.3, “BCVG” SFRs (for byte code verification)	Included in section 5.5 of this ST; will be regarded during production of the composite TSAM TOE
[JCSPP] section 5.1.9, “SCPG” SFRs (for smart card platform, i.e. operating system and chip)	Already regarded in terms of SFRs for [JCOP41V231] during the corresponding evaluation, not relevant for the IT environment of the composite TSAM TOE

Environment SFRs from [JCSPP]	Coverage in this ST
[JCSPP] section 5.1.10, “CMGRG” SFRs (for card manager)	Already regarded in terms of SFRs for [JCOP41V231] during the corresponding evaluation, not relevant for the IT environment of the composite TSAM TOE

1532 **Table 8-10: Coverage of IT Environment SFRs from [JCSPP]**

9 Appendix

9.1 Abbreviations

3/DES	Triple Data Encryption Standard
APDU	Application Protocol Data Unit
BCV	Byte Code Verification
BSN	Batch Settlement Number
DES	Data Encryption Standard
EEPROM	Electrically Erasable Programmable Read Only Memory
GP	GlobalPlatform
GPK	GlobalPlatform Key
IC	Integrated Circuit
JC	JavaCard
JCP	JavaCard Platform
LCS	Life Cycle State
MAC	Message Authentication Code
MID	Merchant Identifier
MK	Management Key
OSP	Organizational Security Policy
POS	Point Of Sales
PP	Protection Profile
RC	Retry Counter
ROM	Read Only Memory
SAR	Security Assurance Requirement
SCP	Smart Card Platform
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SO	Security Objective
SOE	Security Objective for the Environment
ST	Security Target
TD	Transaction Data
TID	Terminal Identifier
TMD	Terminal Management Data
TOE	Target of evaluation
TSAM	Terminal Security Access Module
TSN	Transaction Serial Number
TSF	TOE Security Functions
TSP	TOE Security Policy
WK	Working Key

9.2 References

[3/DES]	Federal Information Processing Standard Publication, FIPS PUB 46-3 October 1999.
[AIS20]	Anwendungshinweise und Interpretationen zum Schema, AIS 20: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 1, 02.12.1999, Bundesamt für Sicherheit in der Informationstechnik
[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, August 2005, Version 2.3
[CEM]	Common Evaluation Methodology for Information Technology Security Evaluation, August 2005, Version 2.3
[ISO7816-4]	*ISO/IEC 7816-4, Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Interindustry commands for interchange*
[JCOP41V231]	NXP P541G072V0P (JCOP 41 v2.3.1), Secure Smart Card Controller
[JCOP41V231ST]	*SECURITY TARGET, NXP P541G072V0P (JCOP 41 v2.3.1), Secure Smart Card Controller*, version 2.13, date 2007-06-01, IBM Deutschland Entwicklung GmbH
[JCSPP]	*Java Card System Protection Profile Collection*, Version: 1.0b, August 2003. This Document contains 4 protection profile, whereas "*Java Card System – Minimal Configuration Protection Profile*" (registered at DCSSI under Registration number PP/0303) is relevant for this ST
[PP0002]	*Smartcard IC Platform Protection Profile*, Version 1.0, July 2001 (registered at BSI under Registration number BSI-PP-0002)
[ST0348]	Security Target Lite, BSI-DSZ-CC-0348, Version 1.2, 17.01.2006, *Evaluation of the Philips P5CT072V0P, P5CC072V0P, P5CD072V0P and P5CD036V0P Secure Smart Card Controllers*