MultiApp Platform

Security Target

TABLE OF CONTENTS

Figures

Tables

1 ST introduction

1.1 ST overview

This document is not a complete security target itseft. Indeed, the document describes the security services provided by the platform to a SSCD application. We aim to factorize as much as possible the common part (in particular the platform) of the two SSCD evaluations.

The security target of the IC is described in its turn in [ST_SAMSUNG].

1.2 References

Reference	Title
[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCIMB-2005-08-001, version 2.3, August 2005 (conform to ISO 15408).
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements CCIMB-2005-08-002, version 2.3, August 2005 (conform to ISO 15408).
[CC-3]	Common Criteria for Information Technology security Evaluation Part 3: Security Assurance Requirements CCIMB-2005-08-003, version 2.3, August 2005 (conform to ISO 5408).
[CEM]	Common Methodology for Information Technology Security Evaluation CCIMB-2005-08-004, version 2.3, August 2005.
[PP/JCS]	Java Card System Protection Profile Version 1.0b, August 2003.
[PP/BSI-0002]	Smart Card IC Platform Protection Profile, version 1.0, registered by BSI in 2001 under PP-BSI-0002, Eurosmart document (SSVG Protection Profile).
[ST_SAMSUNG]	Security Target of S3CC91C 16-bit RISC Microcontroller for smart card. Version 1.0, August 2007.
[FIPS 46-3]	FIPS 46-3: DES Data Encryption Standard (DES and TDES). National Institute of Standards and Technology
[FIPS 197]	FIPS 197: AES Advanced Encryption Standard. National Institute of Standards and Technology.
[AIS20]	AIS20, Functional Classes and Evaluation Methodologiy for Deterministic Random Number Generator, version 1, December 1999, BSI.
[RSA PKCS#1]	PKCS #1 v2.1: RSA Cryptography Standard
[FIPS 180-2]	FIPS-46-3: Secure Hash Standard (SHA). National Institute of Standards and Technology.
[ISO 7816-4]	Identification cards - Integrated circuit(s) cards with contacts, Part 4: Interindustry commands for interchange
[ISO 7816-6]	Identification cards - Integrated circuit(s) cards with contacts, Part 6: Interindustry data elements

Reference	Title
[ISO 7816-9]	Identification cards - Integrated circuit(s) cards with contacts, Part 9: Additional Inter industry commands and security attributes.
[ISO 9796-2]	ISO/IEC 9796-2
[JCAPI221]	Application Programming Interface Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 23, 2003
[JCRE221]	Runtime Environment Specification Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 2003
[JCVM221]	Virtual Machine Specification Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 2003
[JCAPN221]	Application Programming Notes for the Java Card™ Platform, Sun Microsystems, Inc, version 2.2.1, October 2003.
[JVM]	The Java Virtual Machine Specification. Lindholm, Yellin. ISBN 0-201-43294-3.
[GP]	Global Platform. Card Specification – v2.1.1, March 2003.

2 TOE Description

Most of the TOE information is provided in the security target of the SSCD applications. In this section, only the platform description is provided.

MultiApp is a Java Open Platform that complies with two major industry standards:

1. Sun's Java Card 2.2.1, which consists of the Java Card 2.2.1 Virtual Machine, Java Card 2.2.1 Runtime Environment and the Java Card 2.2.1 Application Programming Interface.
2. The GlobalPlatform Card Specification version 2.1.

MultiApp contains the following components (see Figure 1):

- The *Native Layer* that provides the basic card functionalities (memory management, I/O management and cryptographic libraries) with native interface with the dedicated IC. The cryptographic library includes TDES, RSA standard and CRT (up to 2048), hashing (SHA-1, SHA-256), OBKG (RSA), and RNG.
- The *Java Card Runtime Environment*, which provides a secure framework for the execution of Java Card programs and data access management (firewall).
- The *Java Card Virtual Machine*, which provides the secure interpretation of bytecodes.
- The *API* including the standard Java Card API, the JCF API (Biometry) and Gemalto proprietary API (SecureAPI, GemUtil, Mifare, CryptoTest).
- The *Open Platform Card Manager*, which provides card, key and application management functions (contents and life-cycle) and security control.

The MultiApp platform provides the following services:

1. Initialization of the Card Manager and management of the card life cycle,
2. Secure installation of the application under Card Manager control,
3. Extradition services to allow several applications to share a dedicated security domain,
4. Deletion of applications under Card Manager control,
5. Secure operation of the applications through the API,
6. Card basic security services as follows:
   - Checking environmental operating conditions using information provided by the IC,
   - Checking life cycle consistency,
   - Ensuring the security of the PIN objects,
   - Generating random number,
   - Handling secure data object and backup mechanisms,
   - Managing memory content,
   - Providing mechanisms to prohibit other applets to interfere with the electronic signature applet.

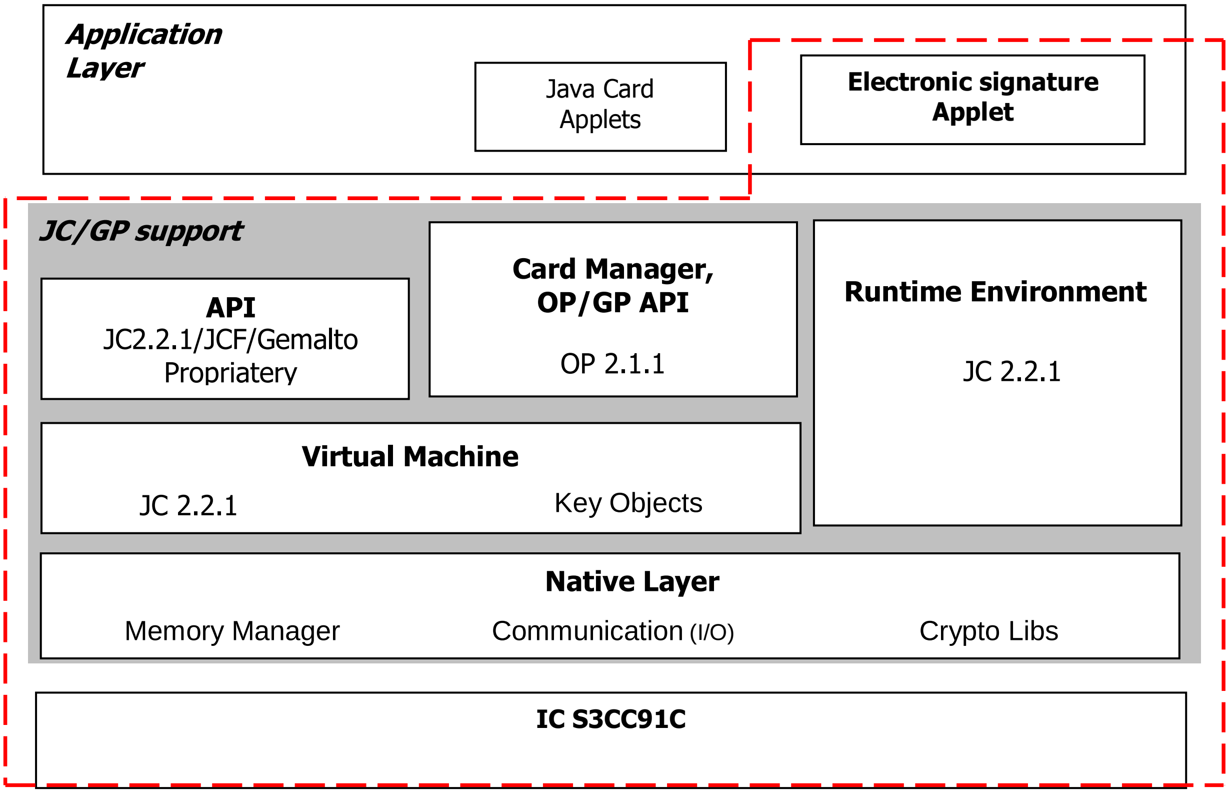

Figure 1. MultiApp platform architecture

3 Conformance claim

The PP claim is presented in the SSCD Security Targets.

4 Security problem definition

4.1 Assets

The platform assets are presented in the following table.

D. CODE	Applets code, the platform code and the corrective softmask (if necessary).
D.GP_REGISTRY	GP registry that contains Card Manager data for card management operations.
D.LOCK_STATE	The availability of applet loading on the card (by the Card Manager or any other entity)
D.ISD_KEYS	ISD keys, *i.e.* the Card Manager keys used during applet initialization and card personalization. Those include keys for authentication, encryption and integrity (MAC).
D.JAVA_OBJECT	Any Java data objects (owned by an application or by the platform)

4.2 Subjects

The platform subjects are presented in the following table:

S.Card_Manager	**Card Issuer**, which manages the card contents and controls application privileges.
S. Package	**Java Card packages** loaded on the platform and acts on behalf of the applet developer.
S.JCRE	**JCRE** acts with the "system priviledge" when accessing to D.JAVA_OBJECT
S.OFFCARD	**Attacker**. A human or process acting on his/her behalf being located outside the TOE. The main goal of the S.OFFCARD attacker is to access application sensitive information. The attacker has a **high level potential attack** and **knows no secret.**

4.3 Threats

The platform threats are presented in the following table:

T.Plt_Integrity	Integrity of the platform data and code. **S.OFFCARD** tries to alter platform stored sensitive data (assets) or code to gain access to unauthorized data or operations. This threat concerns **D.ISD_KEYS**, **D.GP_REGISTRY, D.LOCK_STATE** and **D.CODE.**
T.Plt_Confidentiality	Confidentiality of platform data. **S.OFFCARD** tries to disclose platform-stored data to gain access to unauthorized operations. This threat concerns **D.ISD_KEYS.**
T.Plt_Install	**S.OFFCARD** fraudulently install an applet on the card. This concerns either the installation of an unauthorized applet or an attempt to induce a malfunction in the TOE through the installation process. This threat concerns applets installation and mainly **D.GP_REGISTRY**.

T.Plt_Execution	**S.OFFCARD** or **S.Package** executes code in order to gain illegal access to platform or applet resources. This threat deals with **D.CODE** and **D.JAVA_OBJECT** access.
T.Plt_Operate	**S.OFFCARD** or **S.Package** tries to modify the platform behavior by unauthorized or incorrect use of commands, or by producing malfunction conditions. This includes bad command, authentication bypass, in-secure state by insertion or interruption of session. This threat concerns all platform assets.

4.4 Assumptions

The platform assumption is presented in the following table:

A.Applets	It is assumed that the other instanciable applets (than the electronic signature applet) on the platform are safely installed through the Card Manager and they operate under the Card Manager control. This applies to the applets defined as instanciable in the SSCD security targets

4.5 Organizational Security Policies

The OSPs of the IC are described in its security target [ST_SAMSUNG]. The following table contains the OSP of the platform:

P.Plt_Support	The platform is built with Java Card 2.2.1 and GP 2.1.1 and allows the electronic signature application to operate in a secure environment. The platform support: - Secure electronic signature application installation and extradition, - Secure deletion of the application instantiation, - Secure operating environment with detection of environmental trouble shooting - Secure execution environment and data sharing The Platform shall provide cryptographic services for the electronic signature applets in particular, RSA (up to 2048), TDES, SHA-1, SHA-256, OBKG (RSA), RNG.

5 Security objectives

5.1 Security objectives for the TOE

The security objectives of the IC are described in its security target [ST_SAMSUNG]. The security objectives of the platform are presented in the following table:

OT.Plt_Integrity	The platform shall ensure that the sensitive data (assets) stored in the memory is protected against corruption or unauthorized modification. The platform shall provide means to verify the integrity of its code.
OT.Plt_Confidentiality	The platform shall ensure that the sensitive information is protected against disclosure while being stored or used. The platform shall provide mechanisms to securely manage keys to avoid unauthorized access, disclosure or snooping.
OT.Plt_Reallocation	The platform shall ensure that the re-allocation of a memory block does not disclose the sensitive information previously stored in that block.
OT.Plt_Install	The platform shall ensure that only the authorized administrator is allowed to install/delete applets. The platform must ensure that applet initialization performed under secure conditions.
OT.Plt_Execution	The platform shall ensure that only the authorized administrator is allowed to manage the card content through the dedicated commands.
OT.Plt_Firewall	The platform shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and the TSFs.
OT.Plt_Operate	The platform shall ensure correct operation of its security function and guarantee that the environment, in which the application operates, is safe. The platform shall provide appropriate feedback information upon detection of potential violation.
OT.Plt_Support	The platform is built with Java Card 2.2.1 and GP 2.1.1 and allows the electronic signature application to operate in a secure environment. The platform will support: - Secure electronic signature application installation and extradition, - Secure deletion of electronic signature instantiation. - Secure operating environment with detection of environmental trouble shooting, - Secure execution environment and data sharing The platform shall provide cryptographic services for the electronic signature applications in particular, RSA (up to 2048), TDES, SHA-1, SHA-256, OBKG (RSA), RNG.

5.2 Security objectives for the environment

OE.Applet	Instanciable applets (which are not the electronic signature applet) on the platform shall be safely installed through authorized platform administrator, under the Card Manager control. This applies to the applets defined as instanciable in the SSCD security target.

6 Security requirements

6.1 TOE security functional requirements

[ST_SAMSUNG] deals with the security functional requirements of [PP/BSI-0002]. In this section, we only provide the security functional requirements of th platform.

6.1.1 Platform security functional requirements list

Identification	**DESCRIPTION**
FAU	Security audit
FAU_ARP.1	Security alarms
FAU_SAA.1	Potential violation analysis
FCS	Cryptographic support
FCS_COP.1	Cryptographic operation
FDP	**User data protection**
FDP_ACC.1	Subset Access control
FDP_ACC.2	Complete Access control
FDP_ACF.1	Security attributes based access control
FDP_IFC.1	Subset information flow control
FDP_IFF.1	Simple security attributes
FDP_RIP.1	Subset residual information protection
FDP_SDI.2	Stored data integrity monitoring and action
FDP_UIT.1	Basic data exchange intergrity
FIA	**Identification and Authentication**
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
FIA_USB.1	User-subject binding
FMT	**Security management**
FMT_MOF.1	Management of security function behavior
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attribute initialization
FMT_MTD.1	Management of TSF data
FMT_SMF.1	Specification of Management Function
FMT_SMR.1	Security roles
FPT	**Protection of the TOE Security function**
FPT_RVM.1	Non bypassability of the TSP
FPT_SEP.1	TSF Domain separation
FPT_TDC.1	Inter TSF Basic TSF Data consistency
FTP	**Trusted path/Channel**
FTP_TRP.1	Trusted Path

Table 1. Platform security functional requirements list

6.1.2 FAU Security audits

6.1.2.1 FAU_ARP.1 Security alarms

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

	List of disruptive actions: 1. Reset the card and clear all volatile memory. 2. Block the action that produced the security violation and throw an exception. 3. Terminate the card (put the card life cycle to TERMINATED) and mute 4. Mute the card.

6.1.2.2 FAU_SAA.1 Potential violation analysis

FAU_SAA.1.1	The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.
FAU_SAA.1.2	The TSF shall enforce the following rules for monitoring audited events: a) Accumulation or combination of the following **auditable events** known to indicate a potential security violation: 1. Card Manager life cycle state inconsistency (D.GP_REGISTRY) 2. Integrity errors on D.ISD_KEYS 3. Illegal Access to D.JAVA_OBJECT 4. Unavailability of resources audited through the object allocation mechanism b) Any other rules: **none.**

6.1.3 FCS – Cryptographic support

6.1.3.1 FCS_CKM.1/TDES Cryptographic key generation

FCS_CKM.1.1/ TDES	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **TDES for the generation of session keys** and specified cryptographic key sizes **128 bits, 168 bits** that meet the following standards: **None.**

6.1.3.2 FCS_COP.1/TDES Cryptographic operation

FCS_COP.1.1/ TDES	The TSF shall perform **TDES encryption and decryption** in accordance with a specified cryptographic algorithm **TDES-CBC, TDES-EBC** and cryptographic key sizes **112 bits for TDES 2 keys, 168 bits for TDES 3 keys** that meet the following: **[FIPS 46-3]**.

Application note:

The TOE can also encrypt and decrypt using DES algorithm with 56 bits key, but this is to be considered as a service. The DES algorithm is no longer considered as resistant to high level attacks.

6.1.3.3 FCS_COP.1/RSA Cryptographic operation

FCS_COP.1.1/ RSA	The TSF shall perform **RSA encryption and descryption** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024, 1152, 1280, 1536 or 2048 bits** that meet the following: **standard and CRT**.

6.1.3.4 FCS_COP.1/SHA Cryptographic operation

FCS_COP.1.1/ SHA-1	The TSF shall perform **secure hashing** in accordance with a specified cryptographic algorithm **SHA-1, SHA-256** and cryptographic key sizes **none** that meet the following: **FIPS 180-2**.

Application note:

This cryptographic operation does not use key.

6.1.3.5 FCS_COP.1/RNG Cryptographic operation

FCS_COP.1.1/ RNG	The TSF shall perform **Random Number Generation** in accordance with a specified cryptographic algorithm **Random Number Generator** and cryptographic key sizes **None** that meet the following: **ANSI X9.17 Appendix C.**

Application note:

This cryptographic operation does not use key.

6.1.4 FDP – User data protection

6.1.4.1 FDP_ACC.1 Subset access control

FDP_ACC.1.1/ Card Manager SFP	The TSF shall enforce the **Card Manager SFP** on the following list of subjects, objects and operations.	
Subjects	**Objects**	**Operations**
S.Card_Manager	D.GP_REGISTRY	Applet installation and deletion Change the Card Life Cycle state Change the Application Life Cycle state

6.1.4.2 FDP_ACC.2 Complete access control

FDP_ACC.2.1/ Firewall SFP	The TSF shall enforce the **Frewall SFP** on **S.Package, S.JCRE, D.JAVA_OBJECT** and all operations among subjects and objects covered by the SFP.
Operation	**Description**
OP.ARRAY_ACCESS (D.JAVA_OBJECT, field*)*	Read/Write an array component.
OP.INSTANCE_FIELD (D.JAVA_OBJECT, field*)*	Read/Write a field of an instance of a class in the Java programming language
OP.INVK_VIRTUAL (D.JAVA_OBJECT, method, arg1,...*)*	Invoke a virtual method (either on a class instance or an array object)
OP.INVK_INTERFACE (D.JAVA_OBJECT, method, arg1,...*)*	Invoke an *interface* method.
OP.THROW (D.JAVA_OBJECT)	Throwing of an object (**athrow**).
OP.TYPE_ACCESS (D.JAVA_OBJECT, class*)*	Invoke **checkcast** or **instanceof** on an object.
OP.JAVA (...)	Any access in the sense of [JCRE221], §6.2.8.
*OP.CREATE (*Sharing, LifeTime*)*	Creation of an object (**new** or **makeTransient** call).

Operations (prefixed with " *OP* ") of this policy are described above. Each operation has a specific number of parameters given between brackets, among which there is the "**accessed object** ", the first one, when applicable. Parameters may be seen as security attributes that are under the control of the subject performing the operation. Note that accessing array's components of a **static** array, and more generally fields and methods of **static** objects, is an access to the corresponding *D.JAVA_OBJECT*.

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

6.1.4.3 FDP_ACF.1 Security attributes based access control

FDP_ACF.1.1/ Card_Manager SFP	The TSF shall enforce the **Card Manager SFP** to objects based on **following attributes.**	
Subject/object	**Attribute**	**Values**
S.Card_Manager	Authentication	Yes, No
	Secure Channel	Open, Not Open
D.GP_REGISTRY	Card Life Cycle state	OP_READY, INITIALIZED, SECURED, CARD_BLOCKED, TERMINATED
D.LOCK_STATE	Available Load	Yes, No

FDP_ACF.1.2/ Card_Manager SFP	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed.
Attributes	**Rules**
Authentication Secure Channel Card Life Cycle state Available Load	• The **Secure Channel** is set to Open only if Card Manager has been correctly authenticated and **Authentication [Card Manager]** is set to Yes. • Operations on applications are allowed only if **Card life Cycle state** is set to OP_READY, INITIALIZED or SECURED and if Card Manager has been correctly authenticated with **Authentication [Card Manager]** set to Yes. • Only Card Manager correctly authenticated with **Authentication [Card Manager]** set to Yes is allowed to update **D.GP_REGISTRY** during Applet Install/Delete. • Only Card Manager correctly authenticated with **Authentication [Card Manager]** set to Yes is allowed to set the Applet Life Cycle in **GP_REGISTRY** to INSTALL.

FDP_ACF.1.3/ Card Manager SFP	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**
FDP_ACF.1.4/ Card Manager SFP	The TSF shall explicitly deny access of subjects to objects based on the rule: ▪ **No body** can modify the Available Install attribute of D.ISD_LOCK_STATE.

FDP_ACF.1.1/ Firewall SFP	The TSF shall enforce the **Firewall SFP** to objects based on the following: **(1) the security attributes of the covered subjects and objects, (2) the currently active context and (3) the SELECTed applet** c**ontext**.	
Subject/object	**Attribute**	**Values**
S.JCRE	None	
S.Package	Context	Package AID or "JCRE"
D.JAVA_OBJECT	Context	Package AID or "JCRE"
	Sharing	Standard (both filed sand methods are under firewall policy), or Shareable Interface Object (SIO), or JCRE Entry Point (temporary or permanent), or Global Array
	Lifetime	CLEAR_ON_DESELECT or PERSISTENT
	SELECTed applet context	Package AID or "None"

Both "the currently active context" and "the SELECTed applet context" are internal security attributes to the platform, that is, not attached to any specific object or subject. The currently active context is defined in Section 6.1.2.1 of [JCRE221]. The SELECted applet context is the context of the selected applet and so, must be either a package AID or "None" (when no applet is selected.

FDP_ACF.1.2/ Firewall SFP	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **by the Firewall SFP.**
▪ **R.JAVA.1** ([JCRE221] §6.2.8) An *S.PACKAGE* may freely perform any of *OP.ARRAY_ACCESS*, *OP.INSTANCE_FIELD*, *OP.INVK_VIRTUAL*, *OP.INVK_INTERFACE*, *OP.THROW* or *OP.TYPE_ACCESS* upon any *D.JAVA_OBJECT* whose Sharing attribute has value "JCRE Entry Point" or "Global Array". ▪ **R.JAVA.2** ([JCRE221] §6.2.8) An *S.PACKAGE* may freely perform any of *OP.ARRAY_ACCESS*, *OP.INSTANCE_FIELD*, *OP.INVK_VIRTUAL*, *OP.INVK_INTERFACE* or *OP.THROW* upon any *D.JAVA_OBJECT* whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if *D.JAVA_OBJECT* 's Context attribute has the same value as the active context. ▪ **R.JAVA.3** ([JCRE221] §6.2.8.10) An *S.PACKAGE* may perform *OP.TYPE_ACCESS* upon an *D.JAVA_OBJECT* whose Sharing attribute has value "SIO" only if *D.JAVA_OBJECT* is being cast into (checkcast) or is being verified as being an instance of (instanceof) an *interface* that extends the Shareable *interface*. ▪ **R.JAVA.4** ([JCRE221]§6.2.8.6) An *S.PACKAGE* may perform *OP.INVK_INTERFACE* upon an *D.JAVA_OBJECT* whose Sharing attribute has the value "SIO" only if the invoked *interface* method extends the Shareable *interface*.	

FDP_ACF.1.3/ Firewall SFP	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **The subject *S.JCRE* can freely perform *OP.JAVA(...)* and *OP.CREATE*, with the exception given in *FDP_ACF.1.4/Firewall SFP*.**

FDP_ACF.1.4/ Firewall SFP	The TSF shall explicitly deny access of subjects to objects based on the **rules:** 1. **Any subject with *OP.JAVA* upon an *D.JAVA_OBJECT* whose LifeTime attribute has value "CLEAR_ON_DESELECT" if *D.JAVA_OBJECT* 's Context attribute is not the same as the SELECTed applet Context.** 2. **Any subject with *OP.CREATE* and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the SELECTed applet Context**.

6.1.4.4 FDP_IFC.1 Subset information flow control

FDP_IFC.1.1/JCVM SFP	The TSF shall enforce the **JCVM information flow control SFP** on **the following subjects, information and operations**.
Subject/Information	**Description**
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.MEMBER	Any object's field, static field or array position.
I.DATA	JCVM Reference Data: *objectref addresses of temporary JCRE Entry Point objects and global arrays.*
Operation	**Description**
OP.PUT(S_1, S_2, I)	Transfer a piece of information *I* from S_1 to S_2.

Application note: References of temporary *JCRE entry points*, which cannot be stored in *class* variables, instance variables or array components, are transferred from the internal memory of the

JCRE (TSF data) to some stack through specific APIs (*JCRE* owned exceptions) or *JCRE* invoked methods (such as the **process(APDU apdu)**); these are causes of *OP.PUT(S_1,S_2,I)* operations as well.

6.1.4.5 FDP_IFF.1 Simple security attributes

FDP_IFF.1.1/JCVM SFP	The TSF shall enforce the **JCVM information flow control SFP** based on the following types of subject and information security attributes: **S.LOCAL, S.MEMBER, I.DATA and the currently active context**.

FDP_IFF.1.2/JCVM SFP	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **An operation *OP.PUT(S_1, S.MEMBER, I)* is allowed if and only if the active context is "JCRE"; other *OP.PUT* operations are allowed regardless of the active context's value**.

FDP_IFF.1.3/JCVM SFP	The TSF shall enforce the **none**.

FDP_IFF.1.4/JCVM SFP	The TSF shall provide the following the **none**.

FDP_IFF.1.5/JCVM SFP	The TSF shall explicitly authorise an information flow based on the following rules: **all JCRE Permanent Entry Point Object may be stored in a S.MEMBER**.

FDP_IFF.1.6/JCVM SFP	The TSF shall explicitly deny an information flow based on the following rules: **the storage of the reference of an object with attribute JCRE Temporary Entry Point Object or Global Array in a static field, instance field or array element is forbidden**.

6.1.4.6 FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1/Card Manager SFP	The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of resource** for the following objects: • **D.ISD_KEYS**

6.1.4.7 FDP_SDI.2 Stored data integrity monitoring and action

The following data persistently stored by the TOE have the user attribute "**integrity checked persistent stored data**":

- Keys: D.ISD_KEYS
- Card Life Cycle state and the Applet Life Cycle state (in D.GP_REGISTRY)

FDP_SDI.2.1/ KEYS	The TSF shall monitor user data stored within the TSC for **integrity errors** on all objects, based on the following attributes: **integrity checked persistent stored data.**
FDP_SDI.2.2/ KEYS	Upon detection of a data integrity error, the TSF shall: • **Prohibit the use of the altered data** • **Inform S.Card_Manager about integrity error.** • **Mute the card**

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

	checked persistent stored data.
FDP_SDI.2.2/ Card_life_cycle	Upon detection of a data integrity error, the TSF shall: • **Prohibit the use of the altered data** • **Inform S.Card_Manager about integrity error.** • **Terminate the card**

FDP_SDI.2.1/ Applet_life_cycle	The TSF shall monitor user data stored within the TSC for **integrity errors** on all objects, based on the following attributes: **integrity checked persistent stored data.**
FDP_SDI.2.2/ Applet_life_cycle	Upon detection of a data integrity error, the TSF shall: • **Prohibit the use of the altered data** • **Inform S.Card_Manager about integrity error.** • **Terminate the card**

6.1.4.8 FDP_UIT.1 Data exchange confidentiality

FDP_UIT.1.1	The TSF shall enforce the **Card Manager SFP,** to be able to **transmit** and **receive** objects in a manner protected from **modification** and **insertion** errors.

6.1.5 FIA – Identification and Authentication

6.1.5.1 FIA_ATD.1 User attribute definition

FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: - **Authentication** - **Context (i.e. Package AID)**

6.1.5.2 FIA_UAU.1 Timing of authentication

<table>
<tr><td rowspan="2"><b>FIA_UAU.1.1</b></td><td colspan="2">The TSF shall allow <b>the following TSF mediated actions</b> on behalf of the user to be performed before the user is authenticated.</td></tr>
<tr><td><b>S.Card_Manager</b></td><td><b>Get Data</b>
<b>Select</b>
<b>Initialize Update</b>
<b>Manage Channel</b></td></tr>
<tr><td><b>FIA_UAU.1.2</b></td><td colspan="2">The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.</td></tr>
</table>

6.1.5.3 FIA_UID.1 Timing of identification

FIA_UID.1.1	The TSF shall allow the **selection** of an **application** on behalf of the user to be performed before user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user**.**

6.1.5.4 FIA_USB.1 User-subject binding

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

<u>**Application Note**</u> (dependencies): the security attributes are listed in FIA_ATD.1.

6.1.5.5 FMT_MOF.1 Management of security function behavior

FMT_MOF.1.1	The TSF shall restrict the ability to **modify the behavior of** the functions **listed below** to **S.Card_Manager:** • Delete application • Install application • Update D.ISD.KEY

Application Note: S.Card_Manager may assign a delegated security domain (at the installation of this SD) that represents S.Card_Manager in loading and installating an application.

6.1.5.6 FMT_MSA.1 Management of security attributes

FMT_MSA.1.1/ Card Manager	The TSF shall enforce the **Card Manager SFP** to restrict the ability to **perform the following operations on** the security attributes **defined below** to the ***Card manager.***

Object	Security attribute	Operation	SFP	Role
			See FDP_ACF.1	See FMT_SMR.1
D.GP_REGISTRY	Card Life Cycle state	Modify	Card Manager	Card Manager (phase 6)

Application Note: The delegated application (by S.Card_Manager) has the same role as the Card Manager itself.

FMT_MSA.1.1/ JCRE	The TSF shall enforce the **FIREWALL SFP and the JCVM SFP** to restrict the ability to **modify** the security attributes **the active context and the SELECTed applet Context** to **the JCRE (*S.JCRE*)**.

6.1.5.7 FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1	The TSF shall enforce the **Card Manager SFP,** the **Firewall SFP,** and the **JCVM SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **none** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.8 FMT_MTD.1 Management of TSF data

FMT_MTD.1.1	The TSF shall restrict the ability to **access or modify** the **following TSF data** to **the *Card Manager* role: D.ISD_KEY.**

6.1.5.9 FMT_SMF.1 Specification of Management function

FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: • Delete application • Install application • Update D.ISD.KEY • Modify D.GP_REGISTRY

6.1.5.10 FMT_SMR.1 Security roles

FMT_SMR.1.1	The TSF shall maintain the roles **defined in the following list.** **<u>The roles list</u>:** **1. The Card Manager role (phase 5, 6)** The *Card Manager is the personalizer of the Card and the Card Issuer as there is no*

	post issuance loading of application. *The Card Manager may be* in charge of application install operation and for setting the application state to INSTALLED and SELECTABLE, then for setting the Card Life Cycle state to SECURED. The *Card Manager* may be in charge of deleting an application. **2.The *Application User* role (phase 7).** After application installation, the platform only sees application users. The access to platform resources is granted according to Java Card firewall access conditions.
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

6.1.5.11 FPT_RVM.1 Non-Bypassability of the TSP

FPT_RVM.1	The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

6.1.5.12 FPT_SEP TSF Domain separation

FPT_SEP.1.1	The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1.2	The TSF shall enforce separation between the security domains of subjects in the TSC.

6.1.5.13 FPT_TDC.1 Inter-TSF data consistency

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **the CAP files (shared between the Byte Code Verifier and the TOE), the bytecode and its data arguments (shared between applets and API packages)** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2	The TSF shall use **the following interpretation rules** when interpreting the TSF data from another trusted IT product.
	<u>Interpretation rules list</u>:
	o **The [JCVM221] specification;**
	o **Reference export files;**
	o **The** [**ISO 7816-6] rules;**
	o **The [GP] specification**

6.1.6 FTP – Trusted path/channels

6.1.6.1 FTP_TRP.1 Trusted path

FTP_TRP.1.1	The TSF shall provide a communication path between itself and **local** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.
FTP_TRP.1.2	The TSF shall permit **local users** to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for **D. ISD_KEYS load** and application **Install/Extradite** operation.

6.2 Security requirements for the IT environment

Threre is no security requirement for the IT environment on the platform.

7 TOE summary specification

This section describes the seucurity functions of the platform. The security functions of the IC are described in its security target [ST_SAMSUNG].

7.1 TOE security functions

SF_CARD_AUTHENTICATION	Card authentication
SF_CARD_CRYPTO	Card cryptographic algorithm & key management
SF_CARD_EMANATION	Emanation protection
SF_CARD_INTEGRITY	Card objects integrity
SF_CARD_MGR	Card Manager
SF_CARD_PROTECT	Card operation protection
SF_CARD_SECURE_MESSAGING	Card Secure Messaging

Table 2. TOE security functions provided by the platform

7.1.1 SF_CARD_AUTHENTICATION: Card authentication

This security function ensures the management of the administrator authentication:

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

As the user PIN is managed by the application itself, this security function manages administrator authentication that opens secure channel for communication with the terminal. Authentication failure is managed using a retry counter. When the predefined number of unsuccessful authentication is reached the card will be BLOCKED.

This function is SOF-High.

7.1.2 SF_CARD_CRYPTO: Card cryptographic algorithm and keys managements

This security function provides the cryptographic algorithm and functions used by the TSF:

- <u>TDES algorithm</u> only support 112-bit key and 168-bit key
- <u>RSA algorithm</u> supports 1024-to-2048 bits keys. The RSA algorithm is SW and does not use the IC cryptograhic library. The platform supports both standard and CRT RSA.
- <u>Random generator</u> uses the certified Hardware Random Generator that fulfils the requirements of AIS20 (see [ST_SAMSUNG]).
- <u>SHA-1 and SHA-256 algorithms</u>

This security function controls all the operations relative to the card keys management.

- <u>Key generation</u>: The TOE provides the following:

RSA key generation manages 1024 to 2048-bits long keys. The RSA key generation is SW and does not use the IC cryptographic library.

The TDES key generation (for session keys) uses the random generator.

- <u>Key destruction</u>: the TOE provides a specified cryptographic key destruction method that makes Key unavailable.

This security function ensures the confidentiality of keys during manipulation and ensures the de-allocation of memory after use.

This security function is supported by the IC security function **SF5 (Cryptography)** for Random Number Generator (see [ST_SAMSUNG]).

7.1.3 SF_CARD_EMANATION: Emanation protection

This SF protects the electronic signature application data RAD and SCD against snooping. The SF ensures that:

- The TOE shall not emit electromagnetic radiation in excess of unintelligible emission enabling access to RAD and SCD.
- The TOE shall ensure that the attacker S.OFFCARD is not able to use I/O, VCC or Ground interface to gain access to RAD and SCD.

This security function is supported by the IC security function **SF4 (Hardware countermeasures for unobservability)** (see [ST_SAMSUNG]).

7.1.4 SF_CARD_INTEGRITY: Card objects integrity

This security function provides a means to check the integrity of data stored in EEPROM: the cryptographic keys, including the persistently stored data SCD, RAD and SVD of the electronic signature application, and the card life cycle state.

In case of integrity error detection, this SF will prohibit the use of the altered data, and take appropriate actions: mute or terminate the card.

This SF also ensures that no residual information is available after a PIN update or clearance.

This SF supports SF_CARD_PROTECT by checking platform data integrity before use.

This SF also provides the authorized users with the capability to verify the integrity of stored TSF executable code.

7.1.5 SF_CARD_MGR: Card manager

This security function ensures the administration of the card during its life-cycle: personalization phase, and usage phase. This SF enforces the following access control policies:

- Applet installation, extradition, deletion
- Java objects access control (firewall)

This SF analyzes the incoming commands and checks the access rights, according to the life-cycle and the required secure environment.

This SF ensures that only authorized administrator can manage card contents and manages the following access control policies:

- At the end of the platform initialization, the Card Manager (Issuer) security domain is created and the associated Card Manager keys are loaded before the Card Life Cycle state is set to OP-READY. Once in OP-READY state, the card is under Card Manager control.
- Once the platform is set to OP-READY, applets can be installed by the authenticated Card Manager, through a successfully opened secure channel. Application security domains are created and the associated keys are loaded. Only an authenticated Card Manager is allowed to modify the card life-cycle, lock the load operation, and update the keys. Access to Java objects is controlled by the firewall using the security context attached to each objects.
- During usage phase, the Card Manager controls access to a Java object through the Firewall, using the Security context associated with each object.
- The selection of an application is always allowed.

This security function is dependent on SF_CARD_AUTHENTICATION and SF_CARD_SECURE_MESSAGING.

7.1.6 SF_CARD_PROTECT: Card operation protection

This security function ensures the protection of the TSF and supports the following operations.

- Analyze potential violation on the card life-cycle inconsistency, the PIN and keys integrity error, the illegal access to Java objects, and the unavailability of resources.
- Take action upon violation detection: reset the card, block the action, terminate or mute the card.
- Check start-up security conditions: the consistency of life-cycle, the intergrity of specific data area.
- Check operating conditions periodically by listening the IC sensors.
- Resist to physical attacks (such as out-of-bound voltage, clock frequency and temperature, etc)

In case of error detections this function returns an error or an exception and takes appropriate shield action. If during the TSF execution an unexpected error or an abortion occurs, a secure state will be preserved by resetting security attributes to secure values and if necessary recover the persistently stored data to a secure consistent state.

This security function ensures the atomicity of Java objects update in EEPROM:

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

This SF is supported by the IC security function **SF1 (Environmental security violation recording and reaction)** (see [ST_SAMSUNG]) for attack detecting and resisting.

7.1.7 SF_CARD_ SECURE_MESSAGING: Card secure messaging

This security function ensures the integrity and/or the confidentiality of command/message transmission in a secure channel. The integrity is achieved by adding a message authentication code to the message. The confidentiality is achieved by APDU message data field encryption. These features are used in accordance with the security mode applied to the secure channel.

This SF is activated after a Card Manager authentication that allows the open of the secure channel. This SF ensures that the secure channel is closed after a select application command or in case of error detected within the session.

The secure channel is required for the applet Install or Extradition and Card Manager keys loading (ISD Keys).

This SF depends on SF_CARD_AUTHENTICATION.

MODIFICATION SHEET

Date	Modifications	Author
April 23, 2009	Creating from evaluated ST (V1.5)	Quang-Huy Nguyen

END OF DOCUMENT