

Written by: Aurel Sorin SPORNIC

Usimera Protect

Security Target

PUBLIC VERSION



SIMEOS SECURITY TARGET

CONTENT

1.	ST INTRODUCTION.....	5
1.1	ST IDENTIFICATION.....	5
1.2	ST OVERVIEW	5
1.3	CC CONFORMANCE	6
1.4	REFERENCES	7
2.	TOE DESCRIPTION.....	12
2.1	PRODUCT TYPE	12
2.2	SMART CARD PRODUCT LIFE-CYCLE.....	14
2.3	TOE ENVIRONMENT	16
2.4	TOE INTENDED USAGE	18
3.	TOE SECURITY ENVIRONMENT	19
3.1	ASSETS.....	19
3.2	SUBJECTS.....	21
3.3	ASSUMPTIONS	22
3.4	THREATS.....	22
3.5	ORGANISATIONAL SECURITY POLICIES.....	25
4.	SECURITY OBJECTIVES.....	26
4.1	SECURITY OBJECTIVES FOR THE TOE	26
4.2	SECURITY OBJECTIVES FOR THE ENVIRONMENT	29
5.	IT SECURITY REQUIREMENTS	31
5.1	TOE SECURITY FUNCTIONAL REQUIREMENTS	31
5.2	TOE SECURITY ASSURANCE REQUIREMENTS	61
5.3	SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT	74
6.	TOE SUMMARY SPECIFICATION	80
6.1	TOE SECURITY FUNCTIONS	80
6.2	ASSURANCE MEASURES	83
7.	PP CLAIMS	86
7.1	PP REFERENCE	86
7.2	PP ADDITIONS	86
8.	RATIONALE.....	88
8.1	SECURITY OBJECTIVES RATIONALE	88
8.2	SECURITY REQUIREMENTS RATIONALE	88
8.3	TOE SUMMARY SPECIFICATION RATIONALE	88
	INDEX.....	90

SIMEOS SECURITY TARGET

Figures

Figure 2.1 – The TOE.....	13
Figure 2.2 – Life Cycle.....	14

SIMEOS SECURITY TARGET

1. ST INTRODUCTION

1.1 ST IDENTIFICATION

Title: SIMEOS Security Target (Public version)

Version: 5.0 issued March 23rd, 2007

Registration: Ref. D1019540

Origin: GEMALTO

Commercial name: USIMERA PROTECT

Product ref. T1000230 Usimera Protect 128K crypto on Infineon

The TOE is composed with:

Component	Version number	Supplier
Usimera Protect software	2.1	Gemalto
PSL	0.50.23	Infineon
Chip SLE88CFX4000P	m8830b17	Infineon

1.2 ST OVERVIEW

This Security Target defines the security requirements for the “USIMERA PROTECT” product. This product is of a new generation based on secure Operating System addressing Mobile Communication requirements.

It is a SIM/USIM card based on ETSI Release 5 and Release 6 standards both for SCP and 3GPP. The architecture of the product is an UICC with first level applications, such as SIM – addressing 2G network authentication features and USIM – addressing 3G network authentication features.

The product is compliant with Java Card 2.2.1 and Global Platform 2.1.1, enabling value added services to be installed on card.

The Usimera Protect Security Target focuses on the following security functions:

- **Hardware Tamper Resistance:**
 - This is the chip security layer that meets PP SSVG [PP/BSI-0002].
- **Secure operation of the Java Card Virtual Machine (meet PP JCS):**
 - This is the Java Card Virtual Machine and Operating System that meets [PP/JCS].

The security requirements used from [PP/JCS], Javacard 2.2. standard configuration, are those from the groups: *CoreG* (§5.1.1), *InstG* (§5.1.2), *ADELG* (§5.1.4), *RMIG* (§5.1.5), *LCG* (§5.1.6), *ODELG* (§5.1.7) and *CarG* (§5.1.8).

The IT environment security requirements are the ones defined in the groups *BCVG* (§5.1.3). Due to the extension of the TOE scope compared to its PP definition, the groups *SCPG* (§5.1.9) and *CMGRG* (§5.1.10) became TOE security requirements.
- **2G/3G network authentication:**

SIMEOS SECURITY TARGET

- The card proposes secure authentication through 2G/3G network so that operators' keys and algorithms shall not be compromised.

1.3 CC CONFORMANCE

The compliance is assumed with CC version V2.2 (ISO 15408) ([CC-1], [CC-2], [CC-3]) with Interpretations of January 2004.

This product is a Java Card compliant to the Protection Profile Java Card 2.2 Standard from SUN [PP/JCS] and uses a certified chip conformant to the Protection Profile SSVG (also known as PP/BSI-0002) from Eurosmart, in addition with mobile communication specific functional security requirements, providing the necessary security so that value added applications can be safely loaded and executed on card. Therefore the ST is conformant to [PP/JCS] Protection Profile and IC is conformant to [PP/BSI-0002] Protection Profile.

This ST is CC V2.2 conformant with Part 2.

This ST is CC V2.2 conformant with Part 3 conformant and EAL4 augmented as stated in [PP/JCS], [PP/BSI-0002].

The assurance level for this product is EAL4 augmented by:

- ❑ ADV_IMP.2 (Development – Implementation of the TSF)
- ❑ ALC_DVS.2 (Sufficiency of security measures)
- ❑ AVA_VLA.4 (Vulnerability Assessment – Analyse, Highly resistant)
- ❑ AVA_MSU.3 (Vulnerability Assessment – Analysis and testing for insecure states).

The augmentations may also be found in section §5.2.

The strength level for the TOE security functional requirements is “SOF high” (Strength Of Functions high).

SIMEOS SECURITY TARGET

1.4 REFERENCES

1.4.1 External References [ER]

Reference	Title
[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model January 2004, Version 2.2, Revision 256, CCIMB-2004-01-001.
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements January 2004, Version 2.2, CCIMB-2004-01-002.
[CC-3]	Common Criteria for Information Technology security Evaluation Part 3: Security Assurance Requirements January 2004, Version 2.2, Revision 256, CCIMB-2004-01-003.
[CEM]	Common Methodology for Information Technology Security Evaluation. Evaluation Methodology, January 2004, version 2.2, revision 256, CCIMB-2004-01-004.
[PP/JCS]	Java Card System Protection Profile Version 1.0b issued August 2003, standard 2.2 configuration, SUN document
[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/INFINEON]	Security Target of SLE88CFX4000P/m8830, Version 1.1, Infineon Technologies AG, 20.01.2006
[LDR/INFINEON]	SLE88 Flash Loader Security Concept, Infineon Technologies AG, Rev. 1.1, 2004-03-29.
[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.
[FIPS 180-2]	FIPS-46-3: Secure Hash Standard (SHA). National Institute of Standards and Technology.
[COMP128-2]	Functional Description of COMP128-2 an example algorithm for subscriber authentication and key generation in GSM, GSM Association.
[COMP128-3]	Appendix to be added to the COMP 128-2 specification document to produce COMP128-3. GSM-A document, Reference: SG Doc 114/01, GSM Association.
[ISO 7816-4]	Identification cards - Integrated circuit(s) cards with contacts, Part 4: Interindustry commands for interchange

SIMEOS SECURITY TARGET

Reference	Title
[ISO 7816-6]	Identification cards - Integrated circuit(s) cards with contacts, Part 6: Interindustry data elements
[ISO 7816-9]	Identification cards - Integrated circuit(s) cards with contacts, Part 9: Additional Inter industry commands and security attributes.
[TS 11.11]	Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface, (Release 1999) – v 8.8.0 (2002-09)
[TS 23.040]	Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS); Point to Point (PP) – v 6.0.1 (2002-09)
[TS 23.048]	Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Security Mechanisms for the (U)SIM application toolkit; Stage 2 – v 5.6.0 (2003-03).
[TS 31.102]	3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the USIM Application (Release 5) – v 5.3.0 (2002-12).
[TS 31.111]	3rd Generation Partnership Project; Technical Specification Group Terminals; USIM Application Toolkit (Release 5) – v5.3.0 (2002-12).
[TS 31.115]	3rd Generation Partnership Project; Technical Specification Group Terminals; Secured packet structure for (U)SIM Toolkit applications (Release 6) – v 6.1.0 (2002-09)
[TS 31.116]	3rd Generation Partnership Project; Technical Specification Group Terminals; Remote APDU Structure for (U)SIM Toolkit applications (Release 6) – v 6.1.0 (2002-09)
[TS 31.900]	3rd Generation Partnership Project; Technical Specification Group Terminals; SIM/USIM Internal and External Interworking Aspects – v 5.1.0 (2002-06).
[TS 33.102]	3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 5) – v 5.0.0 (2002-06).
[TS 35.205]	3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General. (Release 5) – v 5.0.0 (2002-06).
[TS 35.206]	3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 5) – V5.1.0 (2003-06).
[TS 43.019]	3rd Generation Partnership Project; Technical Specification Group Terminals; Subscriber Identity Module Application Programming Interface. SIM API for Java Card™ stage 2 – v 5.4.0 (2002-09).
[TS 51.011]	3rd Generation Partnership Project; Technical Specification Group Terminals; Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (Release 5) – v 5.0.0 (2001-12).

SIMEOS SECURITY TARGET

Reference	Title
[TS 51.013]	3rd Generation Partnership Project; Technical Specification Group Terminals; Test specification for SIM API for Java Card™ (Release 4) – v 2.0.0 (2002-09).
[TS 102.221]	Smart cards; UICC-Terminal interface; Physical and logical characteristics (Release 5) – v 5.3.0 (2003-02).
[TS 102.222]	Integrated Circuit Cards (ICC); Administrative commands for telecommunications applications (Release 6) – v 6.0.0 (2003-02).
[TS 102.223]	Smart cards; Card Application Toolkit (CAT) (Release 5) – v5.0.0 (2002-07).
[TS 102.224]	Smart cards; Security mechanisms for UICC based Applications - Functional requirements (Release 6) – v 6.0.0 (2002-04).
[TS 102.225]	Smart cards; Secured packet structure for UICC based applications (Release 6) – v 6.0.0 (2002-04).
[TS 102.226]	Smart Cards; Remote APDU structure for UICC based applications (Release 6) – v 6.3.0 (2003-02).
[TS 102.240]	Smart Cards; UICC Application Programming Interface and Loader Requirements; Service description; (Release 6) – v 6.0.0 (2002-07).
[JCAPI22]	Java Card™ APIs specification version 2.2, Sun Microsystems, Inc.
[JCRE22]	Java Card™ 2.2 Runtime Environment Specification version 2.2, Sun Microsystems, Inc
[JCVM22]	Java Card™ Virtual Machine Specification version 2.2, Sun Microsystems, Inc
[JCAPI221]	Java Card™ APIs specification version 2.2.1, Sun Microsystems, Inc, June 23, 2003.
[JCAPN221]	Application Programming Notes for the Java Card™ Platform, Sun Microsystems, Inc, version 2.2.1, October 2003.
[JCRE221]	Java Card™ Runtime Environment Specification version 2.2.1, Sun Microsystems, Inc, 2003.
[JCVM221]	Java Card™ Virtual Machine Specification version 2.2.1, Sun Microsystems, Inc, 2003.
[JVM]	The Java Virtual Machine Specification. Lindholm, Yellin. ISBN 0-201-43294-3.
[GP]	Global Platform. Card Specification – v2.1.1, March 2003.
[VOP]	Visa Open Platform Card Implementation Requirements Configuration 3 – v2.0

1.4.2 Internal References [IR]

Reference	Title
[DEF_SIMEOS]	SIMEOS Definitions and Acronyms MRD07DEF033000
[FSP_SIMEOS]	SIMEOS Functional Specification D1019561 (FSP_D1019561)

SIMEOS SECURITY TARGET

Reference	Title
[AUT_SIMEOS]	SIMEOS Partial CM automation D1021041 (AUT_D1021041)
[CAP_SIMEOS]	SIMEOS Generation, support and acceptance procedure D1021043 (AUT_D1021043)
[SCP_SIMEOS]	SIMEOS Problem tracking CM coverage D1021044 (SCP_D1021044)
[DEL_SIMEOS]	SIMEOS Detection of modification D1021045 (DEL_D1021045)
[IGS_SIMEOS]	SIMEOS Installation, Generation and Start Up Procedures D1021046 (IGS_D1021046)
[ADM_SIMEOS]	SIMEOS Administrator Guidance D1019545 (ADM_D1019545)
[USR_SIMEOS]	SIMEOS User Guidance D1019543 (USR_D1019543)
[HLD_SIMEOS]	SIMEOS High-level Design (overview document) D1019563 (HLD_D1019563)
[LLD_SIMEOS]	SIMEOS Low-level Design (overview document) D1019566 (LLD_D1019566)
[IMP_SIMEOS]	SIMEOS Implementation representation D1019576 (IMP_D1019576)
[SPM_SIMEOS]	SIMEOS Security Policy Model D1019578 (SPM_D1019578)
[DVS_SIMEOS]	SIMEOS Development Security Documentation D1021035 (DVS_D1021035)
[LCD_SIMEOS]	SIMEOS Life-cycle definition documentation D1021039 (LCD_D1021039)
[TAT_SIMEOS]	SIMEOS Documentation of development tools D1021040 (TAT_D1021040)
[COV_SIMEOS]	SIMEOS Analysis of test coverage D1021030 (COV_D1021030)
[DPT_SIMEOS]	SIMEOS Analysis of the depth of testing D1021031 (DPT_D1021031)
[FUN_SIMEOS]	SIMEOS Test Documentation D1021032 (FUN_D1021032)
[MSU_SIMEOS]	SIMEOS Analysis and testing for insecure states D1021033 (MSU_D1021033)
[SOF_SIMEOS]	SIMEOS Strength of TOE security functions analysis D1021034 (SOF_D1021034)
[VLA_SIMEOS]	SIMEOS Vulnerability Analysis D1019232 (VLA_D1019232)
[PERS_SIMEOS]	SIMEOS Personalisation Manual D1013934
[PT_SIMEOS]	Personalisation Tools for SIMEOS D1021419 (TAT_D1021419)

SIMEOS SECURITY TARGET

1.4.3 Reference of Associated Distribution Sheet

[DS_SIMEOS]	SIMEOS Security Target Distribution sheet D1019542
-------------	--

SIMEOS SECURITY TARGET

2. TOE DESCRIPTION

This part of the ST describes the TOE as an aid to the understanding of its security requirements.

It addresses the product type, the smart card product life cycle, the TOE environment along the smart card life cycle and the general features of the TOE.

2.1 PRODUCT TYPE

The Target of Evaluation (TOE) is the Smart Card Integrated Circuit with Embedded Software in operation and in accordance to its functional specifications.

The TOE is composed of:

- Integrated Circuit including crypto libraries, which is certified separately according to [ST/INFINEON] claiming [PP/BSI-0002],
- Operating System including UICC functionalities [TS 102.221], [TS 102.222],
- Card Manager & Open Platform functionalities [GP].
- Java Card Virtual Machine 2.2.1 [JCVM22], Java Card Runtime Environment 2.2.1 [JCRE22] and Java Card API 2.2.1 [JCAPI221]. The TOE follows the Java Card 2.2 Standard configuration according to [PP/JCS].
- Applications: SIM [TS 11.11], USIM [TS 31.102][TS 31.111] and OTA [TS 23.040][TS 23.048].

The TSF are composed of:

- The chip and its crypto library,
- The GEOS operating system,
- The authentication for SIM and USIM applications,
- The Card Manager loader (applets and packages loading),
- The Java Card System.

In the following chapters the Mobile Communications and Java Card are separated in specific groups. As this ST is a composition, each chapter indicates the items covered by the IC security target.

Next picture represents the TOE, which doesn't include the applets that may be loaded post-issuance.

SIMEOS SECURITY TARGET

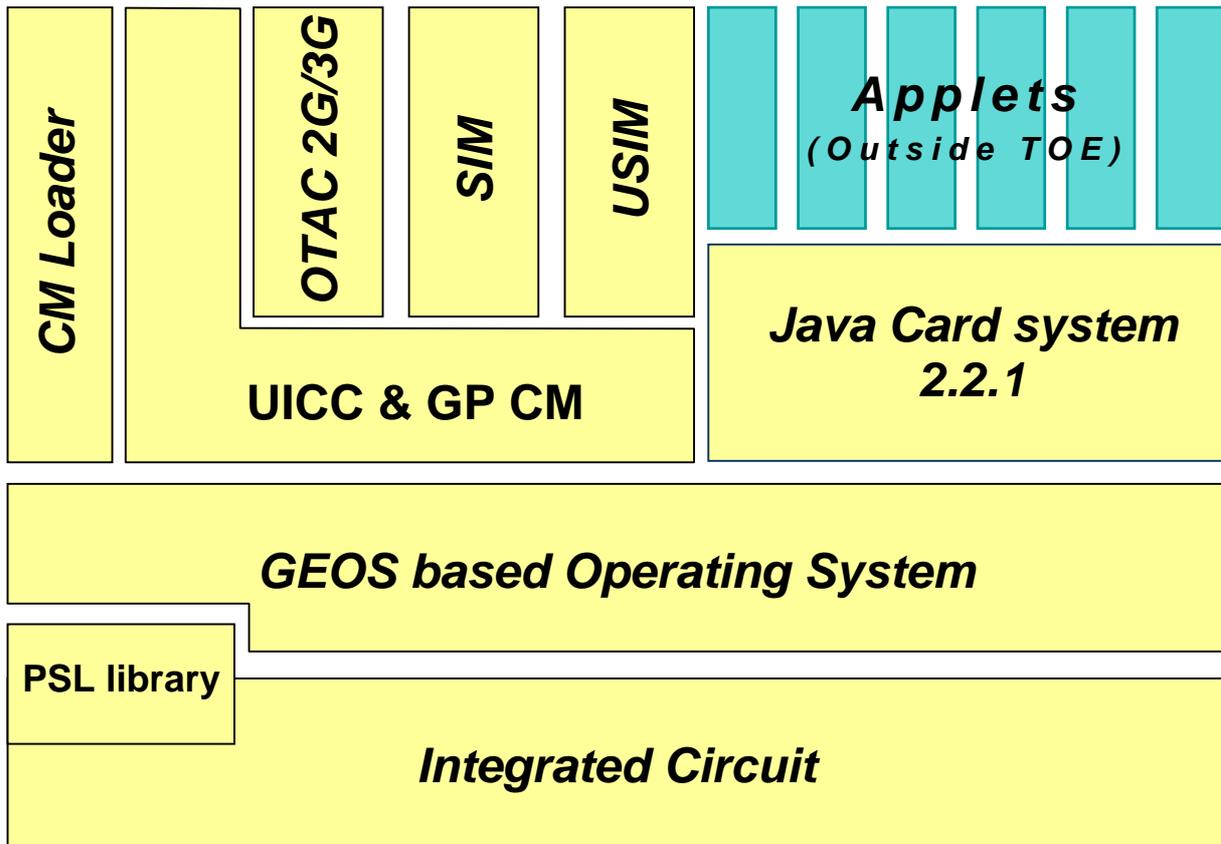


Figure 2.1 – The TOE

SIMEOS SECURITY TARGET

2.2 SMART CARD PRODUCT LIFE-CYCLE

Usimera Protect product life cycle is described in the following picture:

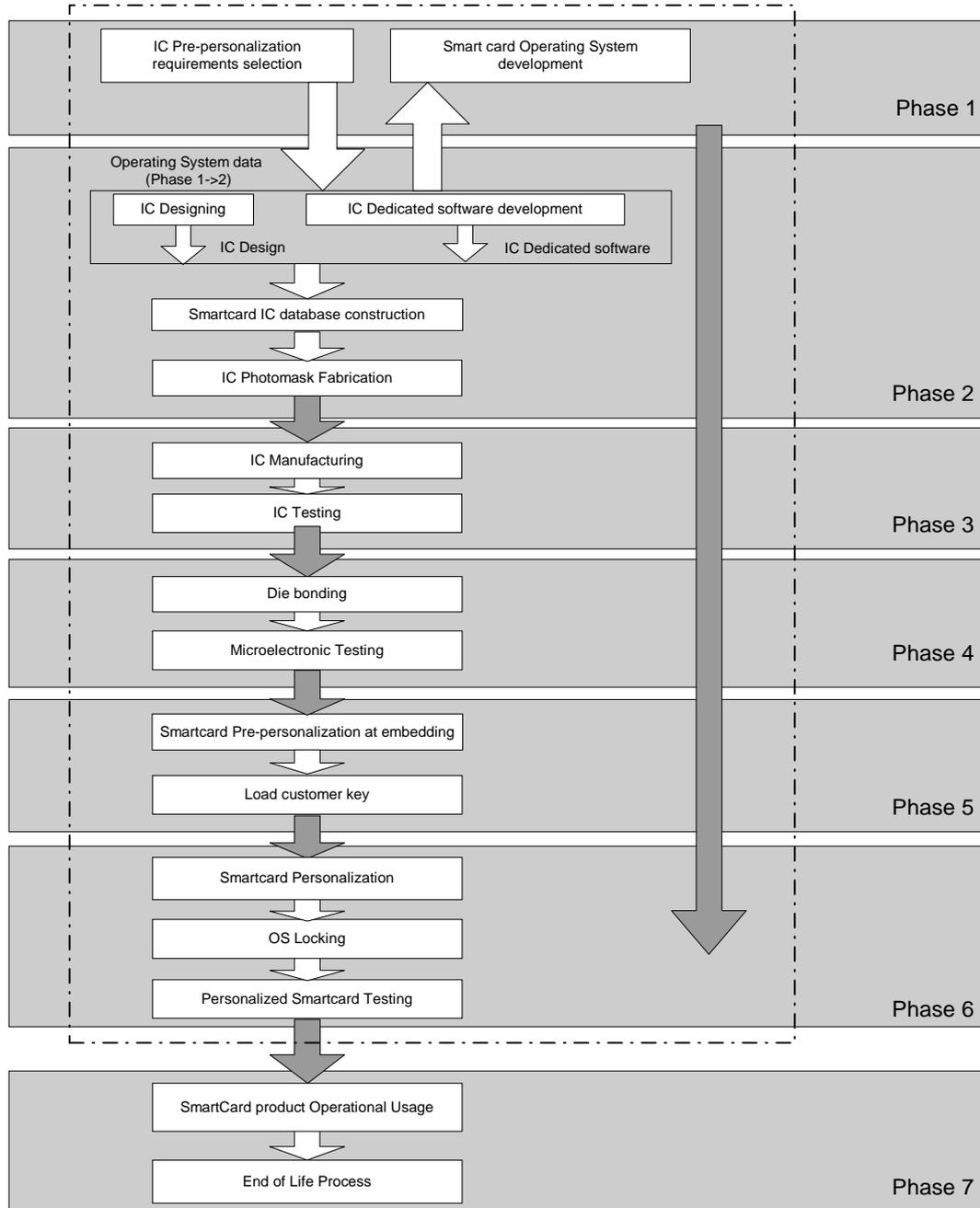


Figure 2.2 – Life Cycle

SIMEOS SECURITY TARGET

The TOE is the product at the end of phase 6 as shown in Figure 2.2. This is the operational Usimera Protect product, as a personalized smart card. As far as the EAL4+ evaluation scope is concerned, phases 1 to 6 are considered as development and manufacturing phases of the product but the TOE is the result of these phases that can consequently be seen as phases of the TOE generation. Phase 7 is the operational phase of the TOE.

The following table gives a description of the product life cycle and explains where the authorities are involved:

Phase 1	Smart Card OS development	The Embedded Software developer is responsible for the development of the Operating System and the specification of initialisation requirements (OS options).
Phase 2	IC design, IC database construction and IC photomask fabrication (IC & DS development)	The IC manufacturer is responsible for these operations, taking as an input the embedded software data given by Embedded Software developer.
Phase 3	IC manufacturing and testing	The IC manufacturer is responsible for producing and testing the IC.
Phase 4	Die bonding. Microelectronic testing.	The IC packaging manufacturer is responsible for die bonding and microelectronic testing. The transport key is changed to Gemalto key.
Phase 5	Pre-personalization (Embedding)	The Pre-personalizer is responsible to store customer loader key encrypted with Gemalto loader key.
Phase 6	Smart card Personalization	The Personaliser is responsible for the Smart Card personalization (change customer key with OS key, loading the OS code and data using the Infineon Loader, locking the capability of code downloading) and for final tests.
Phase 7	Smart card operational-usage	This is the operational usage of the product in the Mobile Communication environment, including the over the air customisation of the product by the end-usage administrator.

SIMEOS SECURITY TARGET

2.3 TOE ENVIRONMENT

Considering the TOE, the environment is defined as follows:

- Development environment corresponding to phases 1 and 2;
- Production and Personalization environments corresponding to phases 3 to 6:
 - Manufacturing environment including the IC test operations, IC packaging, testing and pre-personalization (phases 3 to 5),
 - Personalization environment corresponding to the loading by the Infineon Loader of the OS in the flash memory, personalization and testing of the Smart Card with the user data (phase 6).
- User environment corresponding to the card use by a subscriber on a 2G or 3G network (phase 7).

2.3.1 TOE Development Environment

The TOE described in this ST is developed in different places as indicated below:

IC design	Infineon
Software Design	Gemalto (Axalto S.A., Montrouge and Louveciennes)

In order to ensure security, the environment in which the development takes place must be made secure with access control tracing entries. Furthermore, it is important that all authorised personnel feels involved and fully understands the importance and the rigid implementation of the defined security procedures.

The development begins with the TOE specification. All parties in contact with sensitive information are required to abide by Non-disclosure Agreement.

Design and development of the OS then follows. The engineers use a secure computer system (preventing unauthorised access) to make the conception, design and implementation and to test performances.

Storage of sensitive documents, databases on tapes, CD/DVD and printed circuit layout information are in appropriately locked cupboards/safe. The disposal of unwanted data (complete electronic erasures) and documents (e.g. shredding) is also of paramount importance.

2.3.2 TOE Production and Personalization Environment

The TOE described in this ST is produced in different places as indicated below:

IC manufacturing and Testing	Infineon
IC die bonding	Gemalto (Axalto S.A., Pont Audemer)
Pre-personalization	Gemalto (Axalto S.A., Pont Audemer)
Personalization	Gemalto (Axalto S.A., Pont Audemer)

Testing, programming and deliveries of the TOE then take place. When these are done offsite, they must be transported and worked inside a secure environment with accountability and traceability of all (good and bad) products.

SIMEOS SECURITY TARGET

During the electronic transfer of sensitive data, procedures are established by UPSI (Unified Personalisation Security Infrastructure) to ensure that data arrives only at the destination and it is not accessible at intermediate stages (e.g. stored on a buffer server where system administrators make backup copies). It must also be ensured that transfer is done without modification or alteration.

During fabrication, phases 3 and 4, all the persons involved in storage and transportation operations should fully understand the importance of the defined security procedures.

Moreover, the environment in which these operations take place must be secured.

The TOE Initialisation is performed in [Infineon phase 3; Gemalto phases 4 & 5].

The smart card pre-personalisation takes place in the initialisation environment of the TOE (phase 5).

During smart card pre-personalisation the protection keys are changed from Gemalto keys as set in phase 4 to a customer loader key. This client specific key is encrypted with the Gemalto key when loaded (see key changes in [LDR/INFINEON]).

Initialisation may require a secure environment, with guarantees for the integrity and confidentiality of all operations. By default the 3G Personalisation process insures a good level of security.

Further details on phase 6 will be given by Personalisation specifications [PERS_SIMEOS], [PT_SIMEOS].

2.3.3 TOE User Environment

Smart Cards are used in a wide range of applications to assure authorized conditional access. This specific product is to be used on terminals such as GSM and UMTS handsets or smart card readers.

The end-user environment therefore covers an unprotected environment, thus making it difficult to avoid any abuse of the TOE. The product is prepared accordingly to mitigate such attacks in this environment.

The TOE is ready to be used by the end-user only in phase 7.

2.3.4 The actors and roles

The actors are split in:

Product Developers

The IC designer and DS developer designs the chip and its Dedicated Software (DS). Here it is INFINEON.

The Embedded Software Developer designs the Operating System (OS) according to IC and DS specifications. Here it is GEMALTO.

Product Manufacturer

The IC manufacturer -or founder- designs the photomask, manufactures the IC with the Dedicated Software(PSL). Here the founder and IC manufacturer are INFINEON.

The IC die bounding manufacturer is responsible for the die bounding from the ICs provided by the founder. For this product, the die bounding manufacturer is GEMALTO.

The Smart Card product manufacturer (or Card manufacturer) is responsible to obtain a pre-personalized card from a packaged IC.

For this product, the Smart Card product manufacturer is GEMALTO.

SIMEOS SECURITY TARGET

Personalizer

The Smart Card Personalizer personalizes the card by loading two categories of data:

1. the code and data (OS) belonging to the Developer and Manufacturer of the Card ;
2. the Cardholder data as well as cryptographic keys and PINs.

The Personalizer may also load card issuer applets during this phase. Here it is GEMALTO.

Phase	Administrator	Environment
6	Smart Card Personalizer	Production Environment

The Mobile Communications applications are activated during this phase. At the end of this phase, only applets may be loaded. The card is issued in OP_SECURED state.

Card Issuer

The Card Issuer -short named "issuer"- is the Mobile Communication cellular network operator (GSM or UMTS network). The operator issues cards to its customers who are the "Cardholders". The card belongs to the Card Issuer and its role is to administrate the card in the End Usage Phase. Therefore, the Card Issuer is responsible for selecting and managing the personalization, for managing applets (load, install and delete) as well as for distribution and invalidation of the card.

End User

The Cardholder is a customer of the Card Issuer. The card is personalized with the Cardholder identification and secrets. It is usually the final user.

The roles (administration and usage) are defined in the following tables. During the delivery between phases the responsibility is transferred from the current phase administrator to the next phase administrator.

Phase	Administrator	Environment
7	Card Issuer	Usage Environment

Phase	User	Environment
7	End User	Usage Environment

2.4 TOE INTENDED USAGE

Usimera Protect is a 2G and 3G product, which is compliant with the Release 5 and Release 6 of ETSI Mobile Communication standards. Depending on the handset and network capabilities, it can be used as a USIM card, a SIM card, or both. This product can also support Java Card applets that need security like DES/TDES authentication applets to enable secure access to banking services.

M-Commerce, trusted projects are typical use cases.

SIMEOS SECURITY TARGET

3. TOE SECURITY ENVIRONMENT

3.1 ASSETS

The assets of the TOE are those defined in [PP/JCS], completed by GSM/UMTS ones. The assets of [PP/BSI-0002] are studied in [ST/INFINEON].

3.1.1 User Data

3.1.1.1 Mob Com

D.NET_AUTH_KEY

Network authentication keys, including 3G diversification ones, to secure access to the 2G or 3G Network and associated critical parameters. SIM and Network, or USIM and Network respectively, use this key(s) through Comp128V2, Comp128V3, GSM-Milenage / Milenage algorithm(s) to perform SIM / USIM account authentication. These keys need to be protected in confidentiality and integrity.

D.NET_ACCESS

Access of an authorised account to the network.

3.1.1.2 Java Card

D.APP_CODE

The code of the *applets* and libraries loaded on the card.
To be protected from unauthorized modification.

D.APP_C_DATA

Confidential sensitive data of the applications, like the data contained in an object, a static field of a *package*, a local variable of the currently executed method, or a position of the operand stack.
To be protected from unauthorized disclosure.

D.APP_I_DATA

Integrity sensitive data of the applications, like the data contained in an object, a static field of a *package*, a local variable of the currently executed method, or a position of the operand stack.
To be protected from unauthorized modification.

D.PIN

Any end-user's PIN.
To be protected from unauthorized disclosure and modification.

D.APP_KEYS

Cryptographic keys owned by the *applets*.
To be protected from unauthorized disclosure and modification.

SIMEOS SECURITY TARGET

3.1.2 TSF Data

3.1.2.1 Mob Com

D.AUTH-APPLI_ATTRIB

All SIM and USIM parameters used for Authentication. To be protected in integrity.

D.AUTH-KEY_ATTRIB

Counters and algorithms attributes used for network synchronisation or to prevent replay of authentication data. Contributes to authentication keys protection.

D.AUTH-CRYPTO_ALGO

Cryptographic algorithms used in runtime cryptographic computations for Mobile Communications. This includes 2G and 3G network authentication algorithms.

To be protected from:

- o unauthorised disclosure: only proprietary algorithms COMP128 v2 and v3;
- o unauthorised modification;
- o unauthorised execution.

Application note:

Algorithms available as JC API or used by GP, as TDES and AES are treated in the OS and JC parts.

D.AUTH_CODE

Authentication code. To be protected from unauthorised disclosure.

3.1.2.2 Java Card

D.JCS_CODE

The code of the *Java Card System*.

To be protected from unauthorized disclosure and modification.

D.JCS_DATA

The internal runtime data areas necessary for the execution of the *JVM*, such as, for instance, the stack frame, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures.

To be protected from monopolization and unauthorized disclosure or modification.

D.SEC_DATA

The runtime security data of the *JCRE*, like, for instance, the *AIDs* used to identify the installed *applets*, the *Currently selected applet*, the *current context* of execution and the owner of each object.

To be protected from unauthorized disclosure and modification.

D.API_DATA

Private data of the API, like the contents of its private fields.

To be protected from unauthorized disclosure and modification.

D.JCS_KEYS

Cryptographic keys used when loading a file into the card.

To be protected from unauthorized disclosure and modification.

SIMEOS SECURITY TARGET

D.CRYPTO

Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key.
To be protected from unauthorized disclosure and modification.

3.2 SUBJECTS

3.2.1 Mob Com

S.User_Account

User account granting access to a given mobile communications network (2G or 3G).

S.Operator

Mobile communication operator. It owns the card or at least the keys stored inside.

3.2.2 Java Card

The main *subjects* of the Java Card part are the following ones:

S.PACKAGE

Packages used on the Java Card platform that act on behalf of the applet Developer. These subjects are involved in the **FIREWALL security policy** defined in [SPM] and they should be understood as instances of the subject *S.PACKAGE*.

S.JCRE

The *JCRE*, which acts on behalf of the card issuer. This subject is involved in several of the security policies defined in [PP/JCS] and is always represented by the subject *S.JCRE*.

S.BCV

The bytecode verifier (*BCV*), which acts on behalf of the verification authority. This subject is involved in the **PACKAGE LOADING security policy** defined in Security Functional Requirements and is represented by the subject *S.BCV*.

Application note:

The bytecode verifier is off-card.

S.CRD

The *installer*, which acts on behalf of the card issuer. This subject is involved in the loading of *packages* and installation of *applets*. It could play the role of the on-card entity in charge of package loading, which is involved in the **PACKAGE LOADING security policy** defined in CarG Security Functional Requirements and is represented by the subject *S.CRD*.

Application note:

The *installer* is on-card.

S.ADEL

The *applet deletion manager*, which also acts on behalf of the card issuer. This subject is involved in the **ADEL security policy** defined in applet Deletion Manager Policy.

S.CAD

The *CAD* is involved in the **JCRMI security policy** defined in JCRMI Policy.

SIMEOS SECURITY TARGET

S.SPY

A special subject is involved in the **PACKAGE LOADING security policy**, which acts as the entity that may potentially intercept, modify, or permute the messages exchanged between the verification authority and the on-card entity in charge of package loading.

With the exception of *packages*, the other subjects have special privileges and play key roles in the security policies of the TOE.

3.3 ASSUMPTIONS

The Assumptions of the TOE are based on those defined in [PP/JCS]. The assumptions described in [PP/BSI-0002] are studied in [ST/INFINEON].

3.3.1 Mob Com

A.NET-OP_KEY_MGT

It is assumed that the 2G/3G systems operators apply a high level of confidentiality to card manager and 2G/3G network authentication keys.

It is assumed that the security of the sensitive data stored / handled by the system (Terminals, communication...) is maintained.

3.3.2 Java Card

These Assumptions are those described in [PP/JCS] except *A.NATIVE*. *A.NATIVE* is just removed because API belongs to the TOE and the assumptions described in *A.NATIVE* have been transferred to *OSP.NATIVE*.

A.VERIFICATION

All the bytecodes are verified at least once, before the loading, in order to ensure that each bytecode is valid at execution time.

A.APPLET

Applets loaded post-issuance do not contain native methods. The Java Card specification explicitly "does not include support for native methods" ([JCV22], §3.3) outside the API.

3.4 THREATS

The TOE as defined in chapter 2 is required to counter the threats described hereafter.

A threat agent wishes to abuse the assets either by functional attacks or by environmental manipulation, by specific hardware manipulation, by a combination of hardware and software manipulations or by any other type of attacks. The threats of the TOE are those defined in [PP/JCS] completed by threats specific to Mobile Communications. The threats of [PP/BSI-0002] are included in the evaluation but are studied in [ST/INFINEON].

3.4.1 Common

T.PHYSICAL

The attacker **discloses** or **modifies** the design of the TOE, its sensitive data or application code by physical (opposed to logical) tampering means.

SIMEOS SECURITY TARGET

This threat includes IC failure analysis, electrical probing, unexpected tearing, [Note: extension of the original text from the PP "and DP analysis"] fault injection, side channel attacks using Power and Electromagnetic analysis.

That also includes the modification of the runtime execution of *Java Card System* or *SCP* software through alteration of the intended execution order of (set of) instructions through physical tampering techniques.

3.4.2 Mob Com

T.AUTH-DATA_DISCLOSURE

An attacker may disclose authentication data by logical means (buffer overflow when reading, context changing, usurp read access conditions...).

T.AUTH-DATA_MODIFICATION

Authentication data modification, meaning changing to known values through logical means (similar to authentication data disclosure).

T.AUTH-ALGO_DISCLOSURE

Authentication algorithms disclosure through unauthorised use of 2G or 3G authentication functions (logical attacks including brute force attacks).

T.AUTH-ALGO_MODIFICATION

Cryptographic algorithms (network authentication) deterministic modification through logical means as buffer overflow when writing, context changing, usurp write access conditions, etc.

Application note:

The same result of an incorrect functional code execution in logical attacks may be obtained by physical attacks like fault injection in CPU execution or TSF data, but this is covered by *T.PHYSICAL*.

T.AUTH-ALGO_EXE

Unauthorised execution of the authentication algorithms, to set up statistical or brute force attacks.

T.UNAUTH_NET_ACCESS

Unauthorised access of an account to the mobile network.

3.4.3 Java Card

3.4.3.1 Confidentiality

T.CONFID-JCS-CODE

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

T.CONFID-APPLI-DATA

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

T.CONFID-JCS-DATA

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

SIMEOS SECURITY TARGET

3.4.3.2 Integrity

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.INTEG-APPLI-CODE.2

The attacker modifies (part of) its own or another application code when an application *package* is transmitted to the card for installation.

T.INTEG-APPLI-DATA.2

The attacker modifies (part of) the initialisation data contained in an application *package* when the package is transmitted to the card for installation.

3.4.3.3 Identity Usurpation

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.

3.4.3.4 Unauthorized Executions

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* **executes** a **native method** to bypass a security function such as the *firewall*.

T.EXE-CODE-REMOTE

The attacker performs an unauthorized **remote execution** of a **method** from the *CAD*.

3.4.3.5 Denial of Service

T.RESOURCES

SIMEOS SECURITY TARGET

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

3.4.3.6 Modifications of the Set of Applications

T.INSTALL

The attacker fraudulently **installs** post-issuance an *applet* on the card. This concerns either the installation of an unverified *applet* or an attempt to induce a malfunction in the TOE through the installation process.

3.4.3.7 Card Management

T.DELETION

The attacker **deletes** an *applet* or a *package* already in use on the card, or uses the deletion functions to pave the way for further attacks (putting the TOE in an insecure state).

3.4.3.8 Services

T.OBJ-DELETION

The attacker keeps a reference to a garbage collected object in order to force the TOE to execute an unavailable method, to make it to crash, or to gain access to a memory containing data that is now being used by another application.

3.5 ORGANISATIONAL SECURITY POLICIES

The Organisational security policies described in [PP/BSI-0002] are studied in [ST/INFINEON].

3.5.1 Mob Com

OSP.KeyMgt_DEV_MobCom

Procedures shall ensure that cryptographic keys and relevant security data are used in a secure manner during TOE construction.

3.5.2 Java Card

OSP.VERIFICATION

This policy shall ensure the adequacy between the export files used in the verification and those used for installing the verified file. The policy must also ensure that no modification of the file is performed in between its verification and the signing by the verification authority.

OSP.NATIVE

Those parts of the APIs written in native code as well as any pre-issuance native application on the card shall be conformant with the TOE so as to ensure that security policies and objectives described herein are not violated.

Application note:

Note 1. This was an assumption in the [PP/JCS]. It is most likely to be seen as an OSP as the native applications are included in the TOE and evaluated with.

Note 2. *#.NATIVE* from [PP/JCS] is reduced as follows: No untrusted native code may reside on the card. Loading of native code is forbidden.

SIMEOS SECURITY TARGET

4. SECURITY OBJECTIVES

4.1 SECURITY OBJECTIVES FOR THE TOE

The security objectives stated in [PP/BSI-0002] can be found in [ST/INFINEON] but are covered by this evaluation.

4.1.1 Mob Com

OT.AUTH-KEYS_CONFID

The TOE shall ensure that the 2G or 3G keys and security relevant assets are protected against unauthorised disclosure.

OT.AUTH-KEYS_INTEG

The TOE shall ensure that the 2G, 3G keys and security relevant assets are protected against unauthorised modifications.

OT.AUTH-ALGO_CONFID

The TOE shall ensure that the 2G authentication proprietary algorithms COMP128 v2 and v3 are protected against unauthorised disclosure.

OT.AUTH-ALGO_INTEG

The TOE shall ensure that the cryptographic authentication algorithms are protected against unauthorised modification.

OT.AUTH-ALGO_EXEC

The TOE shall ensure that the cryptographic authentication algorithms are protected against unauthorised (repeated) execution.

OT.ACCOUNT_NET_ACCESS

The TOE shall ensure the prevention of an unauthorised network access.

4.1.2 Java Card

4.1.2.1 Identification

OT.SID

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

4.1.2.2 Execution

OT.OPERATE

The TOE must ensure continued correct operation of its security functions.

OT.RESOURCES

SIMEOS SECURITY TARGET

The TOE controls the availability of resources for the applications.

Application note:

Here the resources include those of the *S.JCRE*.

OT.FIREWALL

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

Application note:

By security domain here it is intended "execution space", which should not be confused with other meanings of "security domains".

OT.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.

OT.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.

Application note:

To be made unavailable means to be physically erased with a default value. Except for local variables that do not correspond to method parameters, the default values to be used are specified in [JCVM22].

OT.SHRD_VAR_CONFID

The TOE shall ensure that any data container that is shared by all applications (like the input/output buffer or a public global variable of the API) 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.

OT.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.

4.1.2.3 Services

OT.ALARM

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

OT.TRANSACTION

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

Application note:

1. Transactions are provided to *applets* as Java Card class libraries.
2. See also the *OT.KEY-MNGT* Application note.

OT.CIPHER

The TOE shall provide 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.

SIMEOS SECURITY TARGET

Application note:

See also the *OT.KEY-MNGT* Application note.

OT.PIN-MNGT

The TOE shall provide means to securely manage PIN objects.

Application note:

1. PIN objects may play key roles in the security architecture of client applications. The way they are stored and managed in the memory of the smart card must be carefully considered, and this applies to the whole object rather than the sole value of the PIN. For instance, the try counter's value is as much sensitive as that of the PIN.
2. See also the *OT.KEY-MNGT* Application note.

OT.KEY-MNGT

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

Application note:

OT.KEY-MNGT, *OT.PIN-MNGT*, *OT.TRANSACTION* and *OT.CIPHER* are actually provided to *applets* in the form of Java Card API's.

OT.REMOTE

The TOE shall provide means to restrict remote access from the *CAD* to the services implemented by the applets on the card. This particularly concerns the *RMI* services introduced in version 2.2 of the Java Card platform.

4.1.2.4 *Applet Management*

OT.INSTALL

The TOE shall ensure that the installation of an *applet* is safe.

OT.LOAD

The TOE shall ensure that the loading of a *package* into the card is safe.

Application note:

Usurpation of identity resulting from a malicious installation of an applet on the card may also be the result of perturbing the communication channel linking the *CAD* and the card. Even if the *CAD* is placed in a secure environment, the attacker may try to capture, duplicate, permute or modify the *packages* sent to the card. He may also try to send one of its own applications as if it came from the card issuer. Thus, this objective is intended to ensure the integrity and authenticity of loaded *CAP files*.

OT.DELETION

The TOE shall ensure that both *applet* and *package* deletion are safe.

4.1.2.5 *Object Deletion*

OT.OBJ-DELETION

The TOE shall ensure the object deletion shall not break references to objects.

4.1.2.6 *IC and Native*

These objectives were Environment objectives into [PP/JCS]. They become Objectives for the TOE because the TOE in this ST includes the SCP.

SIMEOS SECURITY TARGET

OE.NATIVE is renamed to *OT.NATIVE.2* in order to avoid a duplication of names.

OT.NATIVE.2

Those parts of the APIs written in native code as well as any pre-issuance native application on the card shall be conformant with the TOE so as to ensure that security policies and objectives described herein are not violated.

OT.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 (*#.SCP.1* from [PP/JCS]).

OT.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.

OT.SCP.IC

The *SCP* shall possess IC security features.

Application note:

The TOE must be resistant to physical attack and prevent use of security relevant assets derived from these attacks.

OT.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.

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

4.2.1 Common

OE.DEVELOPMENT

During phases 1 to 6, procedures shall be used suitably to maintain the integrity and confidentiality of the assets of the TOE.

4.2.2 Mob Com

OE.KEY_MANAGEMENT

The GSM/UMTS system operator shall apply a high level of confidentiality to administrative and GSM/UMTS keys, so they can be used to authenticate the roles.

4.2.3 Java Card

OE.APPLET

No *applet* loaded post-issuance contains native methods.

SIMEOS SECURITY TARGET

OE.VERIFICATION

Any byte-code must be verified prior to being loaded, in order to ensure that each bytecode is valid at execution time.

The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component is part of the TOE, but outside Java Card, and enforces some of its security functions. Typically the card manager is in charge of the life cycle of the whole card, as well as that of the installed applications (*applets*). The card manager should prevent that card content management (loading, installation, deletion) is carried out, for instance, at invalid states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.

SIMEOS SECURITY TARGET

5. IT SECURITY REQUIREMENTS

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

[ST/Infineon] deals with the security functional requirements of [PP/BSI-0002].

5.1.1 Common

5.1.1.1 Smart card platform (SCPG)

This category includes functional requirements that belonged previously to the IT environment into [PP/JCS]. The Smart Card Platform is now part of the TOE; therefore they are now TOE security functional requirements. The following modifications were done with respect to [PP/JCS]:

- *FPT_RCV.3/SCP* has been merged into *FPT_RCV.3/JCS*;
- *FPT_RVM.1/SCP* has integrated *FPT_RVM.1*.

FPT_AMT.1/SCP Abstract machine testing

FPT_AMT.1.1/SCP The TSF shall run a suite of tests **at each Power-on** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_FLS.1/SCP Failure with preservation of secure state

FPT_FLS.1.1/SCP The TSF shall preserve a secure state when the following types of failures occur: **same as in FPT_FLS.1/JCS**.

FRU_FLT.1/SCP Degraded fault tolerance

FRU_FLT.1.1/SCP [Editorially Refined] The TSF shall ensure **all operations but package loading, applet installation and extension (need the creation of a new chained container)** when the following failures occur: **not enough memory left**.

FPT_PHP.3/SCP Resistance to physical attack

FPT_PHP.3.1/SCP [Editorially Refined] The TSF shall resist to the **physical attacks** on the TOE by responding automatically such that the TSP is not violated.

SIMEOS SECURITY TARGET

FPT_SEP.1/SCP TSF domain separation

FPT_SEP.1.1/SCP 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/SCP The TSF shall enforce separation between the security domains of subjects in the TSC.

Application note:

The use of "security domain" here refers to execution space, and should not be confused with other meanings of security domains.

FPT_RVM.1/SCP Non-bypassability of the TSP

FPT_RVM.1.1/SCP The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Application note:

This component supports *OT.SCP.SUPPORT*, which in turn contributes to the secure operation of the TOE, by ensuring that these latter as well as the supporting platform security mechanisms cannot be bypassed.

FPT_RCV.4/SCP Function recovery

FPT_RCV.4.1/SCP The TSF shall ensure that **reading from and writing to static and objects' fields interrupted by power loss** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

5.1.1.2 Card manager (CMGRG)

This category includes functional requirements that belonged previously to the IT environment into [PP/JCS]. The Card Manager is now part of the TOE; therefore they are now TOE security functional requirements. The following items are missing from the original [PP/JCS] group:

- *FMT_SMR.1/CMGR*; all Security Roles have been merged into *FMT_SMR.1/JCS*.
- *FIA_UID.1/CMGR*; the actions on behalf of the user have been merged into *FIA_UID.1/JCS*.

FDP_ACC.1/CMGR Subset access control

FDP_ACC.1.1/CMGR The TSF shall enforce the **CARD CONTENT MANAGEMENT access control SFP** on **loading of code and keys by the Operator**.

FDP_ACF.1/CMGR Security attribute based access control

FDP_ACF.1.1/CMGR The TSF shall enforce the **CARD CONTENT MANAGEMENT access control SFP** to objects based on the following:

SIMEOS SECURITY TARGET

- o **Subjects:** Byte Code Verifier, Operator
- o **Objects:** applets, keys and PINs.
- o **Security Attributes:** DAP for applets; type and KEK for keys.

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

- o **the Byte Code Verifier loads applets into the card**
- o **the Operator deletes applets in the card.**

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

FDP_ACF.1.4/CMGR The TSF shall explicitly deny access of subjects to objects based on the **none**.

FMT_MSA.1/CMGR Management of security attributes

FMT_MSA.1.1/CMGR The TSF shall enforce the **CARD CONTENT MANAGEMENT access control SFP** to restrict the ability to **modify** the security attributes **code category** to **none**.

FMT_MSA.3/CMGR Static attribute initialisation

FMT_MSA.3.1/CMGR The TSF shall enforce the **CARD CONTENT MANAGEMENT access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/CMGR The TSF shall allow the **none** to specify alternative initial values to override the default values when an object or information is created.

5.1.2 Mob Com

FCS_COP.1/COMP128V2 Cryptographic operation

FCS_COP.1.1/COMP128V2 The TSF shall perform **authentication to network** in accordance with a specified cryptographic algorithm **COMP128V2** and cryptographic key sizes **16 bytes** that meet the following: [**COMP128-2**].

FCS_COP.1/COMP128V3 Cryptographic operation

FCS_COP.1.1/COMP128V3 The TSF shall perform **authentication to network** in accordance with a specified cryptographic algorithm **COMP128V3** and cryptographic key sizes **16 bytes** that meet the following: [**COMP128-3**].

SIMEOS SECURITY TARGET

FCS_COP.1/MILENAGE2G Cryptographic operation

FCS_COP.1.1/MILENAGE2G The TSF shall perform **authentication to network** in accordance with a specified cryptographic algorithm **Milenage 2G** and cryptographic key sizes **16 bytes** that meet the following: [TS 35.206].

Application note:

The key has also diversification parameters.

FCS_COP.1/MILENAGE Cryptographic operation

FCS_COP.1.1/MILENAGE The TSF shall perform **authentication to network** in accordance with a specified cryptographic algorithm **Milenage** and cryptographic key sizes **16 bytes** that meet the following: [TS 35.206].

Application note:

The key has also diversification parameters.

FDP_ACC.1/Auth Subset access control

FDP_ACC.1.1/Auth The TSF shall enforce the **Mobile Communications Access Control Policy on Authentication of an account to the network**.

FDP_ACF.1/Auth Security attribute based access control

FDP_ACF.1.1/Auth The TSF shall enforce the **Mobile Communications access control SFP** to objects based on the following: **the Authentication Key and algorithm**.

Non editorial refinement:

The authentication key has a related attribute: either an authentication counter in 2G or a wrong MAC counter for 3G requests.

FDP_ACF.1.2/Auth The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **an account authenticates itself to the network with a defined association of Authentication Key and algorithm**.

FDP_ACF.1.3/Auth The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

- **an account can authenticate itself upon request from the 2G network. The number of requests accepted is limited.**
- **an account can authenticate itself upon request from the 3G network, if the request is recognised as valid.**

FDP_ACF.1.4/Auth The TSF shall explicitly deny access of subjects to objects based on the

- **value of the 2G authentication counter. Any supplementary request after the upper limit of the counter has expired must be rejected.**

SIMEOS SECURITY TARGET

- o value of the 3G MAC counter. An invalid request must be rejected or any request sent after the wrong MAC counter for requests has expired.

In both cases the account with one of these non-valid attributes cannot authenticate itself to the network.

Global refinement:

A key is associated with an algorithm at its creation time in the manufacturing process. It is not possible to modify it later.

FDP_SDI.2/Auth Stored data integrity monitoring and action

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

FDP_SDI.2.2/Auth Upon detection of a data integrity error, the TSF shall **warn the connected entity**.

Non editorial refinement:

The following data persistently stored by TOE have the user data attribute "integrity checked stored data": Authentication keys and algorithms.

FIA_UID.1/Auth Timing of identification

FIA_UID.1.1/Auth The TSF shall allow **none** on behalf of the user to be performed before the user is identified.

Non editorial refinement:

This restriction applies only to the Mobile Communication actions, therefore only GSM/UMTS actions are not allowed to be done before identification. Different requirements may be specified for other groups (ex. Java Card).

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

Application note:

The user is the network account. The identification is done with the IMSI, stored by the card file system.

FMT_MSA.1/Auth Management of security attributes

FMT_MSA.1.1/Auth The TSF shall enforce the **Mobile Communications SFP** to restrict the ability to **modify** the security attributes **Key not blocked to SIM or USIM on behalf of the User Account**.

FMT_MSA.2/Auth Secure security attributes

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

SIMEOS SECURITY TARGET

FMT_MSA.3/Auth Static attribute initialisation

FMT_MSA.3.1/Auth The TSF shall enforce the **Mobile Communications access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Auth The TSF shall allow the **following roles: none** to specify alternative initial values to override the default values when an object or information is created.

Global refinement:

The initial values are created at the manufacturing stage and cannot be modified afterward. They are assumed to be valid.

FMT_SMR.1/Auth Security roles

FMT_SMR.1.1/Auth The TSF shall maintain the roles **User Account**.

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

FPR_UNO.1/Auth Unobservability

FPR_UNO.1.1/Auth The TSF shall ensure that **anybody** are unable to observe the operation **internal transfer and usage** on **User Data and TSF Data** by the TSF.

Global refinement:

User Data concerned is *D.NET_AUTH_KEY* and TSF Data is *D.AUTH-CRYPTO_ALGO*.

FPT_FLS.1/Auth Failure with preservation of secure state

FPT_FLS.1.1/Auth The TSF shall preserve a secure state when the following types of failures occur: **authentication related TSF data integrity error**.

Global refinement:

The secure state is defined as a state in which the TSF guarantees that no asset will be disclosed.

Application note:

Other failures like power shortage, over voltage, low clock frequency, high clock frequency or unexpected exception are treated by *FPT_FLS.1/SCP*.

FPT_ITT.1/Auth Basic internal TSF data transfer protection

FPT_ITT.1.1/Auth The TSF shall protect TSF data from **modification and disclosure** when it is transmitted between separate parts of the TOE.

SIMEOS SECURITY TARGET

FPT_ITT.3/Auth TSF data integrity monitoring

FPT_ITT.3.1/Auth The TSF shall be able to detect **modification of data** for TSF data transmitted between separate parts of the TOE.

FPT_ITT.3.2/Auth Upon detection of a data integrity error, the TSF shall take the following actions: **list of actions defined in the table hereafter:**

<i>Integrity error type</i>	<i>Action</i>
Unauthorised modification of key attributes	The key is made unavailable
Unauthorised modification of application attributes	The application is made unavailable

5.1.3 Java Card

5.1.3.1 Additional needed SFR

This category contains a requirement that is not listed by [PP/SUN], but that is needed as a dependency for already included SFRs. [PP/SUN] seems not to be complete.

FMT_SMF.1/CM Specification of management functions

FMT_SMF.1.1/CM The TSF shall be capable of performing the following security management functions:

- o **the selection of applets and opening of logical channels;**
- o **the loading and the installing of the applets, with their DAP and AID by the Operator.**

Application note:

This requirement satisfies the dependencies with the *FMT_MSA.1* requirements.

5.1.3.2 Merged SFRs

This group contains SFRs that have been merged from different requirements that [PP/JCS] originally placed into several groups.

FIA_UID.1/JCS Timing of identification

FIA_UID.1.1/JCS The TSF shall allow **JCAPI with already installed applets** on behalf of the user to be performed before the user is identified.

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

Application note:

This SFR was built with *FIA_UID.1/CM* and *FIA_UID.1/CMGR*.

SIMEOS SECURITY TARGET

By users here it must be understood the ones associated to the *packages* (or *applets*), which act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected *applet* or the *package* that is the subject's owner. Means of identification are provided during the loading procedure of the *package* and the registration of *applet* instances.

FMT_SMR.1/JCS Security roles

FMT_SMR.1.1/JCS The TSF shall maintain the roles **JCRE, Operator, Card Manager, applet (RMIG)**.

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

Application note:

1. This SFR combines all *FMT_SMR.1* from [PP/JCS] except the BCV requirement:

- *FMT_SMR.1/JCRE*;
- *FMT_SMR.1/Installer*;
- *FMT_SMR.1/ADEL*;
- *FMT_SMR.1/JCRMI*
- *FMT_SMR.1/CM*,
- *FMT_SMR.1/CMGR*.

2. The Operator includes the *CAD* and the Bytecode Verifier (*FMT_SMR.1/BCV*).

The Card Manager (*FMT_SMR.1/CMGR*) includes the Installer (*FMT_SMR.1/Installer*) and the Applet Deletion Manager (*FMT_SMR.1/ADEL*).

FPT_RCV.3/JCS Automated recovery without undue loss

FPT_RCV.3.1/JCS When automated recovery from **a failure** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2/JCS For **Applet loading, installation and deletion failure; Sensitive data loading failure**, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3/JCS The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **any security attribute** for loss of TSF data or objects within the TSC.

FPT_RCV.3.4/JCS The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

Application note:

1. This SFR combines *FPT_RCV.3/SCP* and *FPT_RCV.3/Installer* from [PP/JCS].

2. The TOE has no maintenance mode. Therefore all operations must be interrupted.

3. The secure state is the mute state.

4. The actual recovery services are provided by the Operating System.

SIMEOS SECURITY TARGET

5.1.3.3 Core (CoreG)

This group includes the Security Functional Requirements as described by the PP, except *FMT_SMR.1/JCRE*. All Security Roles have been merged into *FMT_SMR.1/JCS*.

5.1.3.3.1 Firewall Policy

The following requirements are modified by and therefore included in *LCG*:

- *FDP_ACC.2/FIREWALL*;
- *FDP_ACF.1/FIREWALL*;
- *FMT_MSA.1/JCRE*. *FPT_SEP.1* was integrated into *FPT_SEP.1/SCP*, as now the Smart Card platform is part of the TOE.

FDP_IFC.1/JCVM Subset information flow control

FDP_IFC.1.1/JCVM The TSF shall enforce the **JCVM information flow control SFP** on the following subjects, information and operations.

Non editorial refinement:

Subjects (prefixed with an "S") and information (prefixed with an "I") covered by this policy are:

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.

There is a unique operation in this policy:

Operation	Description
OP.PUT(S ₁ , S ₂ , I)	Transfer a piece of information I from S ₁ to S ₂ .

FDP_IFF.1/JCVM Simple security attributes

FDP_IFF.1.1/JCVM 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 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(S₁, 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 The TSF shall enforce the **none**.

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

FDP_IFF.1.5/JCVM 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.**

SIMEOS SECURITY TARGET

FDP_IFF.1.6/JCVM 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.**

FDP_RIP.1/OBJECTS Subset residual information protection

FDP_RIP.1.1/OBJECTS The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** the following objects: **class instances and arrays.**

Application note:

The semantics of the Java programming language requires for any object field and array position to be initialised with default values when the resource is allocated [JVM], §2.5.1.

FMT_MSA.2/JCRE Secure security attributes

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

Application note:

Secure values conform to the following rules:

- The Context attribute of a **.JAVAOBJECT* must correspond to that of an installed applet or be "JCRE".
- An *O.JAVAOBJECT* whose Sharing attribute is *JCRE Entry Point* or *Global Array* necessarily has "JCRE" value for its Context security attribute.
- An *O.JAVAOBJECT* whose Sharing attribute value is *Global Array* necessarily have "array of primitive Java Card type" as *JavaCardClass* security attribute's value.
- Any *O.JAVAOBJECT* whose Sharing attribute value is not "Standard" has a *PERSISTENT-LifeTime* attribute's value.
- Any *O.JAVAOBJECT* whose LifeTime attribute value is not *PERSISTENT* has an array type as *JavaCardClass* attribute's value.

FMT_MSA.3/FIREWALL Static attribute initialisation

FMT_MSA.3.1/FIREWALL The TSF shall enforce the **FIREWALL access control SFP and the JCVM information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/FIREWALL The TSF shall allow the **authorized identified roles: None** to specify alternative initial values to override the default values when an object or information is created.

5.1.3.3.2 Application Programming Interface

The *FDP_RIP.1/TRANSIENT* requirement was substituted by that one included in *LCG*.

SIMEOS SECURITY TARGET

FCS_CKM.1/AES Cryptographic key generation

FCS_CKM.1.1/AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **AES Key generation** and specified cryptographic key sizes **128 bits** that meet the following: **none (random numbers generation)**.

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 Key generation** and specified cryptographic key sizes **112 bits for TDES 2 keys, 168 bits for TDES 3 keys** that meet the following: **none (random numbers generation)**.

FCS_CKM.2/AES Cryptographic key distribution

FCS_CKM.2.1/AES The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **JC API getkey()** that meets the following: **none**.

FCS_CKM.2/TDES Cryptographic key distribution

FCS_CKM.2.1/TDES The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **JC API getkey()** that meets the following: **none**.

FCS_CKM.3/AES Cryptographic key access

FCS_CKM.3.1/AES The TSF shall perform **access to AES keys** in accordance with a specified cryptographic key access method **secure reading in Memory** that meets the following: **none**.

FCS_CKM.3/TDES Cryptographic key access

FCS_CKM.3.1/TDES The TSF shall perform **access to TDES keys** in accordance with a specified cryptographic key access method **Secure reading in Memory** that meets the following: **none**.

FCS_CKM.4/AES Cryptographic key destruction

FCS_CKM.4.1/AES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: **physical irreversible destruction of the stored key value** that meets the following: **no standard**.

SIMEOS SECURITY TARGET

FCS_CKM.4/TDES Cryptographic key destruction

FCS_CKM.4.1/TDES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: **physical irreversible destruction of the stored key value** that meets the following: **no standard**.

FCS_COP.1/AES Cryptographic operation

FCS_COP.1.1/AES The TSF shall perform **AES encryption and decryption** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **128 bits** that meet the following: **[FIPS 197]**.

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. The applet developer must take into account this consideration.

FCS_COP.1/SHA-1 Cryptographic operation

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

Global refinement:

This cryptographic algorithm does not use a key.

FDP_RIP.1/ABORT Subset residual information protection

FDP_RIP.1.1/ABORT The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **any reference to an object instance created during an aborted transaction**.

Global refinement:

The resource is *any reference to an object instance created during an aborted transaction*.

SIMEOS SECURITY TARGET

FDP_RIP.1/APDU Subset residual information protection

FDP_RIP.1.1/APDU The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** the following objects: **the APDU buffer**.

FDP_RIP.1/bArray Subset residual information protection

FDP_RIP.1.1/bArray The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **the bArray object**.

FDP_RIP.1/KEYS Subset residual information protection

FDP_RIP.1.1/KEYS The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **the cryptographic buffer (D.CRYPTO)**.

FDP_ROL.1/FIREWALL Basic rollback

FDP_ROL.1.1/FIREWALL The TSF shall enforce **the FIREWALL access control SFP and the JCVM information flow control SFP** to permit the rollback of the **OP.JAVA, OP.CREATE** on the **O.JAVAOBJECTS**.

FDP_ROL.1.2/FIREWALL The TSF shall permit operations to be rolled back within the **scope of a select(), deselect(), process() or install() call, notwithstanding the restrictions given in [JCRE22], §7.7, within the bounds of the Commit Capacity ([JCRE22], §7.8), and those described in [JCAPI22]**.

5.1.3.3.3 Card Security Management

This category contains the all SFRs from [PP/JCS], but *FPT_RVM.1* that was merged within the corresponding requirement from the **SCPG**.

FAU_ARP.1/JCS Security alarms

FAU_ARP.1.1/JCS The TSF shall take **the following actions: throw an exception, lock the card session or reinitialise the JCS and its data** upon detection of a potential security violation.

Global refinement:

Potential security violation is refined to one of the following events:

- Applet life cycle inconsistency
- Card tearing (unexpected removal of the Card out of the *CAD*) and power failure
- Abortion of a transaction in an unexpected context (see `abortTransaction()`, [JCAPI22] and ([JCRE22], §7.6.2)
- Violation of the Firewall or JCVM SFPs

SIMEOS SECURITY TARGET

- Unavailability of resources
- Array overflow
- Other runtime errors related to applet's failure (like uncaught exceptions)

Application note:

The thrown exceptions and their related events are described in [JCRE22], [JCAPI22], and [JCVM22].

FDP_SDI.2 Stored data integrity monitoring and action

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

Non editorial refinement:

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

1. PINs
2. keys
3. application sensitive data
4. applet code.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall **warn the connected entity**.

FPT_TDC.1 Inter-TSF basic 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 with applets and API packages)**, when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **the following rules:**

- **The [JCVM22] specification;**
- **Reference export files;**
- **The ISO 7816-6 rules;**
- **The [GP] specification**

when interpreting the TSF data from another trusted IT product.

Application note:

This TOE includes the Card Manager. So this SFR is modified and the interpretation consistency is required between the BCV and the TOE instead of the CM and the TOE.

FPT_FLS.1/JCS Failure with preservation of secure state

FPT_FLS.1.1/JCS The TSF shall preserve a secure state when the following types of failures occur: **those associated to the potential security violations described in FAU_ARP.1/JCS**.

SIMEOS SECURITY TARGET

FPR_UNO.1 Unobservability

FPR_UNO.1.1 The TSF shall ensure that **S.SPY** are unable to observe the operation **cryptographic operations / comparisons operations** on **Key values / PIN values** by **S.JCRE, S.Applet**.

FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self tests **during initial start-up** to demonstrate the correct operation of **the TSF**.

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

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

Application note:

"Initial start-up" means at each power on.

5.1.3.3.4 AID Management

FMT_MTD.1/JCRE Management of TSF data

FMT_MTD.1.1/JCRE The TSF shall restrict the ability to **modify** the **list of registered applets' AID** to the **JCRE**.

FMT_MTD.3 Secure TSF data

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for TSF data.

FIA_ATD.1/AID User attribute definition

FIA_ATD.1.1/AID The TSF shall maintain the following list of security attributes belonging to individual users: **the AID and version number of each package, the AID of each registered applet, and whether a registered applet is currently selected for execution ([JCVM22], §6.5)**.

FIA_UID.2/AID User identification before any action

FIA_UID.2.1/AID The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Application note:

SIMEOS SECURITY TARGET

The User here mentioned are those associated to Packages or Applets.

FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on behalf of the user: **the Context of the package.**

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **each S.Package has a different context independent of the one of the JCRE.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **none.**

Application note:

For *S.PACKAGEs*, the Context security attribute plays the role of the appropriate user security attribute; see *FMT_MSA.1.1/JCRE*.

5.1.3.4 Installer (InstG)

This group includes the Security Functional Requirements as described by the [PP/JCS], except:

- *FMT_SMR.1/Installer*, as all Security Roles have been merged into *FMT_SMR.1/JCS*;
- *FPT_RCV.3/Installer* as Automated Recovery without Undue Loss is usually provided by the Operating System, that is no longer located outside the current TOE. Please see *FPT_RCV.3/JCS*.

FDP_ITC.2/Installer Import of user data with security attributes

FDP_ITC.2.1/Installer The TSF shall enforce the **FIREWALL access control SFP** when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.2.2/Installer The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/Installer The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/Installer The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/Installer The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC:

- A package may depend on (import or use data from) other packages already installed. This dependency is explicitly stated in the loaded package in the form of a list of package AIDs. The loading is allowed only if, for each dependent package, its AID attribute is equal to a resident package AID attribute, the major (minor) Version attribute associated to the former is equal (less than or equal) to the major (minor) Version attribute associated to the latter ([JCVM22], §4.5.2). The intent of this rule is to ensure the binary compatibility of the package with those already on the card ([JCVM22], §4.4).**

SIMEOS SECURITY TARGET

FPT_FLS.1/Installer Failure with preservation of secure state

FPT_FLS.1.1/Installer The TSF shall preserve a secure state when the following types of failures occur: **the installer fails to load/install a *package* / *applet* as described in [JCRE22] §11.1.5.**

FRU_RSA.1/Installer Maximum quotas

FRU_RSA.1.1/Installer The TSF shall enforce maximum quotas of the following resources: **imported packages and declared classes, methods and fields** that **subjects** can use **simultaneously**.

Global refinement:

Subjects here are the packages.

Application note:

1. A *package* may import at most 128 packages.
2. A *package* may declare at most 255 classes and interfaces.
3. A *class* can implement a maximum of 128 public or protected instance methods, and a maximum of 128 instance methods with *package* visibility. These limits include inherited methods.
4. A *class* instance can contain a maximum of 255 fields, where an int data type is counted as occupying two fields ([JCVM21], §2.2.4.2).

5.1.3.5 Applet deletion (ADELG)

This group includes the Security Functional Requirements as described by the PP, except *FMT_SMR.1/ADEL*. All Security Roles have been merged into *FMT_SMR.1/JCS*.

5.1.3.5.1 Applet Deletion Manager Policy

FDP_ACC.2/ADEL Complete access control

FDP_ACC.2.1/ADEL The TSF shall enforce the **ADEL access control SFP** on **S.ADEL, O.JAVAOBJECT, O.APPLET and O.CODE_PKG** and all operations among subjects and objects covered by the SFP.

Non editorial refinement:

Subjects (prefixed with an "S") and objects (prefixed with an "O") covered by this policy are:

S.ADEL The *applet deletion manager*. *This subject is unique.*

O.CODE_PKG The code of a *package*, including all linking information. On the Java Card platform, a package is the installation unit.

O.APPLET Any installed *applet*, its code and data.

O.JAVAOBJECT Java class instance or array.

Operations (prefixed with "OP") of this policy are described in the following table:

SIMEOS SECURITY TARGET

Operation	Description
<i>OP.DELETE_APPLET(O.APPLET,...)</i>	Delete an installed <i>applet</i> and its objects, either logically or physically.
<i>OP.DELETE_PKG(O.CODE_PKG,...)</i>	Delete a <i>package</i> , either logically or physically
<i>OP.DELETE_PKG_APPLET(O.CODE_PKG,...)</i>	Delete a <i>package</i> and its installed <i>applets</i> , either logically or physically.

FDP_ACC.2.2/ADEL 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.

FDP_ACF.1/ADEL Security attribute based access control

FDP_ACF.1.1/ADEL The TSF shall enforce the **ADEL access control SFP** to objects based on the following: (1) the security attributes of the covered subjects and objects, (2) the list of AIDs of the applet instances registered on the card, (3) the attribute ResidentPackages, which journals the list of AIDs of the packages already loaded on the card and (4) the attribute ActiveApplets, which is a list of the active applets' AIDs.

Non editorial refinement:

The following table presents the security attributes associated to the subjects/objects under control of the policy:

Subject/Object	Attributes
<i>O.CODE_PKG</i>	<i>package's AID</i> , dependent packages' AIDs, Static References
<i>O.APPLET</i>	Selection state
<i>O.JAVAOBJECT</i>	Owner, Remote

The package's *AID* identifies the package defined in the CAP file.

When an export file is used during preparation of a CAP file, the version numbers and *AIDs* indicated in the export file are recorded in the CAP files ([JVM21], §4.5.2): the dependent packages AIDs attribute allows the retrieval of those identifications.

Static fields of a package may contain references to objects. The Static References attribute records those references.

An applet instance can be in two different selection states: selected or deselected. If the applet is selected (in some logical channel), then in turn it could either be *currently selected* or just *active*. At any time there could be up to four active applet instances, but only one currently selected. This latter is the one that is processing the current command ([JCRE22], §4).

The Owner of an object is either the applet instance that created the object or the package (library) where it has been defined (these latter objects can only be arrays that initialize static fields of the package).

An object is said to be a Remote if it is an instance of a class that directly or indirectly implements the interface `java.rmi.Remote`.

Finally, there are needed security attributes that are not attached to any object or subject of the TSP: (1) the ResidentPackages Versions (or Resident Image, [JVM21], §4.5) and *AIDs*. They describe the packages that are already on the card, (2) the list of registered applet instances and (3) the ActiveApplets security attribute. They are all attributes internal to the VM, that is, not attached to any specific object or subject of the SPM. These attributes are TSF data that play a role in the SPM.

FDP_ACF.1.2/ADEL The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **by the ADEL SFP: see corresponding refinement.**

Non editorial refinement:

The subject of this policy is *S.ADEL*.

SIMEOS SECURITY TARGET

Some basic common specifications are required in order to allow Java Card *applets* and *packages* to be deleted without knowing the implementation details of a particular deletion manager. In particular, this policy introduces a notion of **reachability**, which provides a general means to describe objects that are referenced from a certain *applet* instance or *package*.

In the context of this policy, an object O is reachable if and only if either: (1) the owner of O is a registered *applet* instance A (O is reachable from A), (2) a static field of a loaded *package* P contains a reference to O (O is reachable from P), (3) there exists a valid remote reference to O (O is remote reachable), and (4) there is an object O' that is reachable according to either (1) or (2) or (3) above and O' contains a reference to O (the reachability status of O is that of O').

The following access control rules determine when an operation among controlled subjects and objects is allowed by the policy:

R.JAVA.14 ([JCRE22], §11.3.4.1, **Applet Instance Deletion**). Deletion). The *S.ADEL* may perform *OP.DELETE_APPLET* upon an *O.APPLET* only if, (1) *S.ADEL* is currently selected, (2) *O.APPLET* is deselected and (3) there is no *O.JAVAOBJECT* owned by *O.APPLET* such that either *O.JAVAOBJECT* is reachable from an applet instance distinct from *O.APPLET*, or *O.JAVAOBJECT* is reachable from a package P, or ([JCRE22], §8.5) *O.JAVAOBJECT* is remote reachable.

R.JAVA.15 ([JCRE22], §11.3.4.1, **Multiple Applet Instance Deletion**). The *S.ADEL* may perform *OP.DELETE_APPLET* upon several *O.APPLET* only if, (1) *S.ADEL* is currently selected, (2) every *O.APPLET* being deleted is deselected and (3) there is no *O.JAVAOBJECT* owned by any of the *O.APPLET* being deleted such that either *O.JAVAOBJECT* is reachable from an applet instance distinct from any of those *O.APPLET*, or *O.JAVAOBJECT* is reachable from a package P, or ([JCRE22], §8.5) *O.JAVAOBJECT* is remote reachable.

R.JAVA.16 ([JCRE22], §11.3.4.2, **Applet/Library Package Deletion**). The *S.ADEL* may perform *OP.DELETE_PCKG* upon an *O.CODE_PCKG* only if, (1) *S.ADEL* is currently selected, (2) no reachable *O.JAVAOBJECT*, from a package distinct from *O.CODE_PCKG* that is an instance of a class that belongs to *O.CODE_PCKG* exists on the card and (3) there is no package loaded on the card that depends on *O.CODE_PCKG*.

R.JAVA.17 ([JCRE22], §11.3.4.3, **Applet Package and Contained Instances Deletion**). The *S.ADEL* may perform *OP.DELETE_PCKG_APPLET* upon an *O.CODE_PCKG* only if, (1) *S.ADEL* is currently selected, (2) no reachable *O.JAVAOBJECT*, from a package distinct from *O.CODE_PCKG*, which is an instance of a class that belongs to *O.CODE_PCKG* exists on the card, (3) there is no package loaded on the card that depends on *O.CODE_PCKG* and (4) for every *O.APPLET* of those being deleted it holds that: (i) *O.APPLET* is deselected and (ii) there is no *O.JAVAOBJECT* owned by *O.APPLET* such that either *O.JAVAOBJECT* is reachable from an applet instance not being deleted, or *O.JAVAOBJECT* is reachable from a package not being deleted, or ([JCRE22], §8.5) *O.JAVAOBJECT* is remote reachable.

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

FDP_ACF.1.4/ADEL The TSF shall explicitly deny access of subjects to objects based on the: **any subject but the S.ADEL to O.CODE_PCKG or O.APPLET must not have access for the purpose of deleting it from the card**.

FMT_MSA.1/ADEL Management of security attributes

FMT_MSA.1.1/ADEL The TSF shall enforce the **ADEL access control SFP** to restrict the ability to **modify** the security attributes **ActiveApplets** to **the JCRE**.

SIMEOS SECURITY TARGET

FMT_MSA.3/ADEL Static attribute initialisation

FMT_MSA.3.1/ADEL The TSF shall enforce the **ADEL access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ADEL The TSF shall allow the **following roles: none** to specify alternative initial values to override the default values when an object or information is created.

5.1.3.5.2 Additional Deletion Requirements

FDP_RIP.1/ADEL Subset residual information protection

FDP_RIP.1.1/ADEL The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **applet instances and/or packages when one of the deletion operations in FDP_ACC.2.1/ADEL is performed on them.**

Application note:

Requirements on de-allocation during *applet / package* deletion are described in [JCRE22] §11.3.4.1, §11.3.4.2 and §11.3.4.3.

FPT_FLS.1/ADEL Failure with preservation of secure state

FPT_FLS.1.1/ADEL The TSF shall preserve a secure state when the following types of failures occur: **the applet deletion manager fails to delete a package / applet as described in [JCRE22], §11.3.4.**

5.1.3.6 RMI (RMIG)

This group includes the Security Functional Requirements as described by the PP, except *FMT_SMR.1/JCRMI*. All Security Roles have been merged into *FMT_SMR.1/JCS*.

5.1.3.6.1 JCRMI Policy

FDP_ACC.2/JCRMI Complete access control

FDP_ACC.2.1/JCRMI The TSF shall enforce the **JCRMI access control SFP** on **S.CAD, S.JCRE, O.APPLET, O.REMOTE_OBJ, O.REMOTE_MTHD, O.ROR, O.RMI_SERVICE** and all operations among subjects and objects covered by the SFP.

Non editorial refinement:

Subjects (prefixed with an "S") and objects (prefixed with an "O") covered by this policy are:

S.CAD The *CAD*. In the scope of this policy it represents the actor that requests, by issuing commands to the card, for RMI services.

S.JCRE The *JCRE* is responsible on behalf of the card issuer of the bytecode execution and runtime environment functionalities. In the context of this security policy, the *JCRE* is in charge of the execution of the commands provided to (1) obtain the initial remote reference of an applet instance and (2) perform Remote Method Invocation.

O.APPLET Any installed *applet*, its code and data.

SIMEOS SECURITY TARGET

O.REMOTE_OBJ A remote object is an instance of a class that implements one (or more) remote interfaces. A remote interface is one that extends, directly or indirectly, the interface `java.rmi.Remote` ([JCAPI22]).

O.ROR A remote object reference. It provides information concerning: (i) the identification of a remote object and (ii) the Implementation class of the object or the interfaces implemented by the class of the object. This is the object's information to which the *CAD* can access.

O.REMOTE_MTHD A method of a remote interface.

O.RMI_SERVICE These are instances of the class `javacardx.rmi.RMIService`. They are the objects that actually process the RMI services.

Operations (prefixed with " *OP* ") of this policy are described in the following table:

Operation	Description
<i>OP.GET_ROR(O.APPLET,...)</i>	Retrieves the initial remote object reference of a RMI based <i>applet</i> . This reference is the seed which the <i>CAD</i> client application needs to begin remote method invocations
<i>OP.INVOKE(O.RMI_SERVICE,...)</i>	Request a remote method invocation on the remote object.

FDP_ACC.2.2/JCRMI 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.

FDP_ACF.1/JCRMI Security attribute based access control

FDP_ACF.1.1/JCRMI The TSF shall enforce the **JCRMI access control SFP** to objects based on the following: **(1) the security attributes of the covered subjects and objects, (2) the list of AIDs of the applet instances registered on the card and (3) the attribute ActiveApplets, which is a list of the active applets'AIDs.**

Non editorial refinement:

The following table presents the security attributes associated to the objects under control of the policy:

Object	Attributes
<i>O.APPLET</i>	Package's AID or none
<i>O.REMOTE_OBJ</i>	Owner, class, Identifier, Exported
<i>O.REMOTE_MTHD</i>	Identifier
<i>O.RMI_SERVICE</i>	Owner, Returned References

For **FDP_ACF.1.2: R.JAVA.18** The *S.CAD* may perform *OP.GET_ROR* upon an *O.APPLET* only if *O.APPLET* is the currently selected applet, and there exists an *O.RMI_SERVICE* with a registered initial reference to an *O.REMOTE_OBJ* that is owned by *O.APPLET*.

R.JAVA.19 The *S.JCRE* may perform *OP.INVOKE* upon *O.RMI_SERVICE*, *O.ROR* and *O.REMOTE_MTHD*, only if, *O.ROR* is valid (as defined in [JCRE22], §8.5) and belongs to the value of the attribute Returned References of *O.RMI_SERVICE*, and the attribute Identifier of *O.REMOTE_MTHD* matches one of the remote methods in the class, indicated by the security attribute class, of the *O.REMOTE_OBJECT* to which *O.ROR* makes reference.

FDP_ACF.1.2/JCRMI The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **by the JCRMI SFP: R.JAVA.18, R.JAVA.19.**

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

SIMEOS SECURITY TARGET

FDP_ACF.1.4/JCRMI The TSF shall explicitly deny access of subjects to objects based on the **any subject but S.JCRE to O.REMOTE_OBJ and O.REMOTE_MTHD for the purpose of performing a remote method invocation.**

FDP_IFC.1/JCRMI Subset information flow control

FDP_IFC.1.1/JCRMI The TSF shall enforce the **JCRMI information flow control SFP** on the following **subjects, information and operations (see refinement).**

Non editorial refinement:

Subjects (prefixed with an " S ") and information (prefixed with an " I ") covered by this policy are:

Subject/Information	Description
S.JCRE	As in the Access control policy
S.CAD	As in the Access control policy
I.RORD	Remote object reference descriptors

A remote object reference descriptor provides information concerning: (i) the identification of the remote object and (ii) the implementation class of the object or the interfaces implemented by the class of the object. The descriptor is the only object's information to which the CAD can access.

There is a unique operation in this policy:

Operation	Description
OP.RET_RORD(S.JCRE,S.CAD,I.RORD)	Send a remote object reference descriptor to the CAD.

A remote object reference descriptor is sent from the card to the CAD either as the result of a successful SELECT FILE command ([JCRE22], §8.4.1), and in this case it describes, if any, the initial remote object reference of the selected applet; or as the result of a remote method invocation ([JCRE22], §8.3.5.1).

FDP_IFF.1/JCRMI Simple security attributes

FDP_IFF.1.1/JCRMI The TSF shall enforce the **JCRMI information flow control SFP** based on the following types of subject and information security attributes: **S.JCRE, S.CAD, ExportedInfo.**

Non editorial refinement:

The following table summarizes which security attribute is attributed to which subject/information:

Subject/Information	Attributes
S.JCRE	None
S.CAD	None
I.RORD	ExportedInfo (Boolean value)

The ExportedInfo attribute of an I.RORD indicates whether the O.REMOTE_OBJ which I.RORD identifies is exported or not (as indicated by the security attribute Export of the O.REMOTE_OBJ).

FDP_IFF.1.2/JCRMI 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.RET_RORD(S.JCRE, S.CAD, I.RORD) is permitted only if the attribute ExportedInfo I.RORD has the value "true" ([JCRE22], §8.5).**

FDP_IFF.1.3/JCRMI The TSF shall enforce the **none.**

SIMEOS SECURITY TARGET

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

FDP_IFF.1.5/JCRMI The TSF shall explicitly authorise an information flow based on the following rules:
OP.INVOKE is allowed if a successful OP.GET_ROR operation was previously successfully executed on the O.ROR supplied in OP.INVOKE and if O.ROR has not been revoked.

FDP_IFF.1.6/JCRMI The TSF shall explicitly deny an information flow based on the following rules:
OP.INVOKE is denied if O.ROR supplied is not valid. OP.INVOKE is denied if the remote method identifier supplied with O.ROR is not the one of a method belonging to the remote object referenced by O.ROR.

FMT_MSA.1/JCRMI Management of security attributes

FMT_MSA.1.1/JCRMI The TSF shall enforce the **FIREWALL access control SFP and the JCVM information flow control SFP** to restrict the ability to **modify** the security attributes **ActiveApplets to the JCRE (S.JCRE)**.

Non editorial refinement:

The intent is to have none of the identified roles to have privileges with regards to the modification of the security attributes.

Application note:

This SFR is included in *FMT_MSA.1/JCRE*.

FMT_MSA.1/EXPORT Management of security attributes

FMT_MSA.1.1/EXPORT The TSF shall enforce the **JCRMI access control SFP and the JCRMI information flow control SFP** to restrict the ability to **modify** the security attributes **export of an O.REMOTE_OBJ to its owner**.

Application note:

The Exported status of a remote object can be modified by invoking its methods `export()` and `unexport()`, and only the owner of the object may perform the invocation without raising a `SecurityException` (`javacard.framework.service.CardRemoteObject`). However, even if the owner of the object may provoke the change of the security attribute value, the modification itself could be performed by the JCRE.

FMT_MSA.1/REM_REFS Management of security attributes

FMT_MSA.1.1/REM_REFS The TSF shall enforce the **JCRMI access control SFP and the JCRMI information flow control SFP** to restrict the ability to **modify** the security attributes **Returned References of an O.RMI_SERVICE to its owner**.

SIMEOS SECURITY TARGET

FMT_MSA.3/JCRMI Static attribute initialisation

FMT_MSA.3.1/JCRMI The TSF shall enforce the **JCRMI access control SFP** and the **JCRMI information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/JCRMI The TSF shall allow the **none** to specify alternative initial values to override the default values when an object or information is created.

FMT_REV.1/JCRMI Revocation

FMT_REV.1.1/JCRMI [Editorially Refined] The TSF shall restrict the ability to revoke the **Returned References security attribute of an O.RMI_SERVICE** to the **JCRE**.

FMT_REV.1.2/JCRMI The TSF shall enforce the rules **that determine the lifetime of remote object references**.

Application note:

The rules previously mentioned are described in [JCRE22], §8.5.

5.1.3.7 Logical channels (LCG)

5.1.3.7.1 Firewall Policy

Except for *FDP_ACC.2/FIREWALL*, *FDP_ACF.1/FIREWALL* and *FMT_MSA.1/JCRE* included hereafter, this policy includes unchanged the functional requirements specified in the **FIREWALL access control SFP** of the group *CoreG*.

FDP_ACC.2/FIREWALL Complete access control

FDP_ACC.2.1/FIREWALL The TSF shall enforce the **FIREWALL access control SFP** on **S.PACKAGE**, **S.JCRE**, **O.JAVAOBJECT** and all operations among subjects and objects covered by the SFP.

Non editorial refinement:

Subjects (prefixed with an "S") and objects (prefixed with an "O") covered by this policy are:

Subject, Object	Description
S.PACKAGE	Any <i>package</i> , which is the security unit of the firewall policy.
S.JCRE	The <i>JCRE</i> . This is the process that manages <i>applet</i> selection and de-selection, along with the delivery of <i>APDUs</i> from and to the smart card device. This subject is unique.
O.JAVAOBJECT	Any object. Note that <i>KEYS</i> , <i>PIN</i> , arrays and <i>applet</i> instances are specific objects in the Java programming language.

Operations (prefixed with "OP ") of this policy are described in the following table. 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.

SIMEOS SECURITY TARGET

Operation	Description
<i>OP.ARRAY_ACCESS</i> (<i>O.JAVAOBJECT, field</i>)	Read/Write an array component.
<i>OP.INSTANCE_FIELD</i> (<i>O.JAVAOBJECT, field</i>)	Read/Write a field of an instance of a class in the Java programming language
<i>OP.INVK_VIRTUAL</i> (<i>O.JAVAOBJECT, method, arg1,...</i>)	Invoke a virtual method (either on a class instance or an array object)
<i>OP.INVK_INTERFACE</i> (<i>O.JAVAOBJECT, method, arg1,...</i>)	Invoke an interface method.
<i>OP.THROW</i> (<i>O.JAVAOBJECT</i>)	Throwing of an object (throw).
<i>OP.TYPE_ACCESS</i> (<i>O.JAVAOBJECT, class</i>)	Invoke checkcast or instanceof on an object.
<i>OP.JAVA</i> (...)	Any access in the sense of [JCRE22], §6.2.8. In our formalization, this is one of the preceding operations.
<i>OP.CREATE</i> (<i>Sharing, LifeTime</i>)	Creation of an object (new or makeTransient call).

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 *O.JAVAOBJECT*.

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.

FDP_ACF.1/FIREWALL Security attribute based access control

FDP_ACF.1.1/FIREWALL The TSF shall enforce the **FIREWALL access control SFP** to objects based on the following: (1) the **security attributes of the covered subjects and objects**, (2) the **currently active context**, (3) the **SELECTed applet Context**, and (4) the **attribute ActiveApplets**, which is a list of the **active applets' AIDs**.

Non editorial refinement:

The following table describes which security attributes are attached to which subject/object of our policy:

Subject /Object	Attributes
<i>S.PACKAGE</i>	Selection Status
<i>S.JCRE</i>	None
<i>O.JAVAOBJECT</i>	Sharing, Context, LifeTime
-	ActiveApplets

The following table describes the possible values for each security attribute:

SIMEOS SECURITY TARGET

Name	Description
Context	<i>Package AID</i> or "JCRE"
Selection Status	Multiselectable, Non-multiselectable or "None"
Sharing	Standard, SIO, JCRE Entry Point, or Global Array
LifeTime	CLEAR_ON_DESELECT or PERSISTENT.
SELECTed applet Context	<i>Package AID</i> or "None"
ActiveApplets	List of package's <i>AIDs</i>

The Java Card platform, version 2.2, introduces the possibility for an *applet* instance to be selected on multiple *logical channels* at the same time, or accepting other applets belonging to the same package being selected simultaneously. These applets are referred to as *multiselectable applets*. *Applets* that belong to a same *package* are either all multiselectable or not ([JCV22], §2.2.5). Therefore, the selection mode can be regarded as an attribute of *packages*. No selection mode is defined for a library *package*.

Support for multiple *logical channels* (with multiple selected applet instances) requires a change to the Java Card System, version 2.1.1, concept of *selected applet*. Since more than one applet instance can be selected at the same time, and one applet instance can be selected on different *logical channels* simultaneously, it is necessary to differentiate the state of the applet instances in more detail. An *applet* instance will be considered an *active applet instance* if it is currently selected in at least one logical channel, up to a maximum of four. An *applet* instance is the *currently selected applet* instance only if it is processing the current command. There can only be one currently selected *applet* instance at a given time. ([JCRE22],§4).

The ActiveApplets security attribute is internal to the VM, that is, not attached to any specific object or subject of the SPM. The attribute is TSF data that plays a role in the SPM.

FDP_ACF.1.2/FIREWALL 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 ([JCRE22] §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 *O.JAVAOBJECT* whose Sharing attribute has value "JCRE Entry Point" or "Global Array".

R.JAVA.2 ([JCRE22] §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 *O.JAVAOBJECT* whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if *O.JAVAOBJECT*'s Context attribute has the same value as the active context.

R.JAVA.3 ([JCRE22] §6.2.8.10) An *S.PACKAGE* may perform *OP.TYPE_ACCESS* upon an *O.JAVAOBJECT* whose Sharing attribute has value "SIO" only if *O.JAVAOBJECT* is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.

R.JAVA.5 An *S.PACKAGE* may perform an *OP.CREATE* only if the value of the Sharing parameter is "Standard".

R.JAVA.20 ([JCRE22] §6.2.8.6) An *S.PACKAGE* may perform *OP.INVK_INTERFACE* upon an *O.JAVAOBJECT* whose Sharing attribute has the value "SIO", and whose Context attribute has the value "*Package AID*", only if one of the following applies:

a) The value of the attribute Selection Status of the package whose *AID* is "*Package AID*" is "Multiselectable",

b) The value of the attribute Selection Status of the package whose *AID* is "*Package AID*" is "Non-multiselectable", and either "*Package AID*" is the value of the currently selected applet or otherwise "*Package AID*" does not occur in the attribute Active applets,

and in either of the cases above the invoked *interface* method extends the Shareable *interface*.

SIMEOS SECURITY TARGET

FDP_ACF.1.3/FIREWALL 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.**

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

- 1) Any subject with **OP.JAVA** upon an **O.JAVAOBJECT** whose **LifeTime** attribute has value **"CLEAR_ON_DESELECT"** if **O.JAVAOBJECT**'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**.

FMT_MSA.1/JCRE Management of security attributes

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

Application note:

The modification of the active context, SELECTed applet Context and ActiveApplets security attributes should be performed in accordance with the rules given in [JCRE22], §4 and ([JCV22], §3.4.

5.1.3.7.2 Additional Requirements on Logical Channels

The following element *FDP_RIP.1/TRANSIENT* substitutes the requirement from *CoreG*:

FDP_RIP.1/TRANSIENT Subset residual information protection

FDP_RIP.1.1/TRANSIENT The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **any transient object**.

Application note:

The events that provoke the de-allocation of any transient object are described in [JCRE22], §5.1.

The clearing of **CLEAR_ON_DESELECT** objects is not necessarily performed when the owner of the objects is deselected. In the presence of multiselectable *applet* instances, **CLEAR_ON_DESELECT** memory segments may be attached to *applets* that are active in different logical channels. Multiselectable applet instances within a same *package* must share the transient memory segment if they are concurrently active ([JCRE22], §4.2.

5.1.3.8 Object deletion (ODELG)

FDP_RIP.1/ODEL Subset residual information protection

FDP_RIP.1.1/ODEL The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **the objects owned by the context of an applet instance which triggered the execution of the method javacard.framework.JCSystem.requestObjectDeletion()**.

SIMEOS SECURITY TARGET

Application note:

Freed data resources resulting from the invocation of the method `javacard.framework.JCSystem.requestObjectDeletion()` may be reused. Requirements on *de-allocation* after the invocation of the method are described in [JCAPI22].

FPT_FLS.1/ODEL Failure with preservation of secure state

FPT_FLS.1.1/ODEL The TSF shall preserve a secure state when the following types of failures occur: **the object deletion functions fail to delete all the unreferenced objects owned by the applet that requested the execution of the method.**

5.1.3.9 *Secure carrier (CarG)*

This group includes the Security Functional Requirements as described by the PP, except:

- *FMT_SMR.1/CM*. All Security Roles have been merged into *FMT_SMR.1/JCS*.
- *FIA_UID.1/CM*; the actions on behalf of the user have been merged into *FIA_UID.1/JCS*.

FCO_NRO.2/CM Enforced proof of origin

FCO_NRO.2.1/CM The TSF shall enforce the generation of evidence of origin for transmitted **application packages** at all times.

FCO_NRO.2.2/CM The TSF shall be able to relate the **identity** of the originator of the information, and the **application package** of the information to which the evidence applies.

FCO_NRO.2.3/CM [Editorially Refined] The TSF shall provide a capability to verify the evidence of origin of information to the **recipient** given **that the related cryptographic keys have been loaded using a secure process.**

Non editorial refinement:

The related cryptographic keys are usually TDES keys that serve to load *applets / packages* through a secure channel.

Application note:

If a new application package is received by the card for installation, the card manager shall first check that it actually comes from the verification authority. The verification authority is the entity responsible for bytecode verification (*S.BCV*).

If there are library packages, they are considered to be included into application packages.

FDP_IFC.2/CM Complete information flow control

FDP_IFC.2.1/CM The TSF shall enforce the **PACKAGE LOADING information flow control SFP** on **S.CRD, S.BCV, S.SPY** and all operations that cause that information to flow to and from subjects covered by the SFP.

Non editorial refinement:

SIMEOS SECURITY TARGET

Subjects (prefixed with an "S") covered by this policy are those involved in the reception of an application package by the card through a potentially unsafe communication channel:

Subject	Description
S.BCV	The subject representing who is in charge of the bytecode verification of the packages (also known as the verification authority).
S.CRD	The on-card entity in charge of package downloading.
S.SPY	Any other subject that may potentially intercept, modify, or permute the messages exchanged between the former two subjects.

The operations (prefixed with "OP") that make information to flow between the subjects are those enabling to send a message through and to receive a message from the communication channel linking the card to the outside world. It is assumed that any message sent through the channel as clear text can be read by the attacker. Moreover, the attacker may capture any message sent through the communication channel and send its own messages to the other subjects.

Operation	Description
OP.SEND(M)	A subject sends a message M through the communication channel.
OP.RECEIVE(M)	A subject receives a message M from the communication channel.

The information (prefixed with an "I") controlled by the typing policy is the APDUs exchanged by the subjects through the communication channel linking the card and the CAD. Each of those messages contain part of an application package that is required to be loaded on the card (either S.BCV or S.SPY), as well as any control information used by the subjects in the communication protocol.

Information	Description
I.APDU	Any APDU sent to or from the card through the communication channel.

FDP_IFC.2.2/CM The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

FDP_IFF.1/CM Simple security attributes

FDP_IFF.1.1/CM The TSF shall enforce the **PACKAGE LOADING information flow control SFP** based on the following types of subject and information security attributes:

Subject / Information	Attribute	Value
S.BCV	DAPKey, OPKeys	Valid
S.CRD	DAPKey, OPKeys	Valid
S.SPY	none	none
I.APDU	SecureLevel	None, MAC, ENC

Subject / Object	Attribute	Value
user	role	Operator, Signatory, None
applet	Transfer	OTA, Local
applet	checked	Boolean
DAP Key	OK	Boolean

SIMEOS SECURITY TARGET

FDP_IFF.1.2/CM The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- o **The user with the security attribute role set to Operator can load an applet with the security attribute Transfer set to OTA.**
- o **The user with the security attribute role set to Service Provider can load an applet with the security attribute Transfer set to Local.**
- o **Only applets with the security attribute Checked set to YES can be transferred.**
- o **The DAP key OK security attribute must be set to TRUE to check the integrity and the origin of the applet.**

FDP_IFF.1.3/CM The TSF shall enforce the **None**.

FDP_IFF.1.4/CM The TSF shall provide the following **None**.

FDP_IFF.1.5/CM The TSF shall explicitly authorise an information flow based on the following rules:

- o **The user with the security attribute role set to Operator can load an applet with the security attribute Transfer set to OTA.**
- o **The user with the security attribute role set to Service Provider can load an applet with the security attribute Transfer set to Local.**
- o **The DAP key OK security attribute must be set to TRUE to check the integrity and the origin of the applet.**

FDP_IFF.1.6/CM The TSF shall explicitly deny an information flow based on the following rules: **No user can load an applet with the security attribute Checked set to NO.**

Global refinement:

S.SPY A subject without the correct DAP attributes cannot modify the transmitted information.

FDP_UIT.1/CM Data exchange integrity

FDP_UIT.1.1/CM The TSF shall enforce the **PACKAGE LOADING information flow control SFP** to be able to **receive** user data in a manner protected from **replay, deletion, modification and insertion** errors.

FDP_UIT.1.2/CM The TSF shall be able to determine on receipt of user data, whether **replay, insertion, deletion and modification** have occurred.

Non editorial refinement:

The modification, deletion, insertion, replay apply for some of the pieces of the application sent by the CAD.

FMT_MSA.1/CM Management of security attributes

FMT_MSA.1.1/CM The TSF shall enforce the **PACKAGE LOADING information flow control SFP** to restrict the ability to **modify** the security attributes **AIDs** to **none**.

SIMEOS SECURITY TARGET

FMT_MSA.3/CM Static attribute initialisation

FMT_MSA.3.1/CM The TSF shall enforce the **PACKAGE LOADING information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/CM [Editorially Refined] The TSF shall allow **none** to specify alternative initial values to override the default values when an object or information is created.

FTP_ITC.1/CM Inter-TSF trusted channel

FTP_ITC.1.1/CM 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/CM The TSF shall permit **the remote trusted IT product** to initiate communication via the trusted channel.

Non editorial refinement:

The **remote trusted IT product** extends the *CAD* placed in the card issuer secured environment notion from the [PP/JCS] with the OTA server, belonging to the Network Operator.

FTP_ITC.1.3/CM The TSF shall initiate communication via the trusted channel for **installing a new application package on the card**.

Application note:

There is no dynamic package loading on the Java Card platform. New packages can be installed on the card only on demand of the Card issuer or Network operator.

5.2 TOE SECURITY ASSURANCE REQUIREMENTS

The [PP/BSI-0002] requirements for the IC evaluation are consistent with these requirements.

The security assurance requirement level is EAL4. The EAL is augmented with AVA_MSU.3, AVA_VLA.4, ADV_IMP.2 and ALC_DVS.2.

5.2.1 ACM Configuration management

5.2.1.1 ACM AUT CM automation

ACM_AUT.1 Partial CM automation

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

SIMEOS SECURITY TARGET

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

ACM_AUT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.2 ACM_CAP CM capabilities

ACM_CAP.4 Generation support and acceptance procedures

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

SIMEOS SECURITY TARGET

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.3 ACM_SCP CM scope

ACM_SCP.2 Problem tracking CM coverage

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 ADO Delivery and operation

5.2.2.1 ADO_DEL Delivery

ADO_DEL.2 Detection of modification

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

SIMEOS SECURITY TARGET

5.2.2.2 ADO IGS Installation, generation and start-up

ADO_IGS.1 Installation, generation, and start-up procedures

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 ADV Development

5.2.3.1 ADV FSP Functional specification

ADV_FSP.2 Fully defined external interfaces

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

SIMEOS SECURITY TARGET

5.2.3.2 ADV_HLD High-level design

ADV_HLD.2 Security enforcing high-level design

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.3 ADV_IMP Implementation representation

ADV_IMP.2 Implementation of the TSF

ADV_IMP.2.1D The developer shall provide the implementation representation for the entire TSF.

ADV_IMP.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.2.2C The implementation representation shall be internally consistent.

SIMEOS SECURITY TARGET

ADV_IMP.2.3C The implementation representation shall describe the relationships between all portions of the implementation.

ADV_IMP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.2.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.4 ADV LLD Low-level design

ADV_LLD.1 Descriptive low-level design

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

SIMEOS SECURITY TARGET

5.2.3.5 ADV_RCR Representation correspondence

ADV_RCR.1 Informal correspondence demonstration

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.6 ADV_SPM Security policy modeling

ADV_SPM.1 Informal TOE security policy model

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modelled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modelled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 AGD Guidance documents

5.2.4.1 AGD_ADM Administrator guidance

SIMEOS SECURITY TARGET

AGD_ADM.1 Administrator guidance

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 AGD_USR User guidance

AGD_USR.1 User guidance

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

SIMEOS SECURITY TARGET

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 ALC Life cycle support

5.2.5.1 ALC DVS Development security

ALC_DVS.2 Sufficiency of security measures

ALC_DVS.2.1D The developer shall produce development security documentation.

ALC_DVS.2.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.2.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.2.3C The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

ALC_DVS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.2.2E The evaluator shall confirm that the security measures are being applied.

5.2.5.2 ALC LCD Life cycle definition

ALC_LCD.1 Developer defined life-cycle model

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

SIMEOS SECURITY TARGET

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.3 ALC_TAT Tools and techniques

ALC_TAT.1 Well-defined development tools

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6 ATE Tests

5.2.6.1 ATE_COV Coverage

ATE_COV.2 Analysis of coverage

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

SIMEOS SECURITY TARGET

5.2.6.2 ATE_DPT Depth

ATE_DPT.1 Testing: high-level design

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

ATE_DPT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.3 ATE_FUN Functional tests

ATE_FUN.1 Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.4 ATE_IND Independent testing

ATE_IND.2 Independent testing - sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

SIMEOS SECURITY TARGET

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.7 AVA Vulnerability assessment

5.2.7.1 AVA MSU Misuse

AVA_MSU.3 Analysis and testing for insecure states

AVA_MSU.3.1D The developer shall provide guidance documentation.

AVA_MSU.3.2D The developer shall document an analysis of the guidance documentation.

AVA_MSU.3.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.3.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.3.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.3.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.3.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

AVA_MSU.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.3.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.3.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

SIMEOS SECURITY TARGET

AVA_MSU.3.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_MSU.3.5E The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

5.2.7.2 AVA_SOF Strength of TOE security functions

AVA_SOF.1 Strength of TOE security function evaluation

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.2.7.3 AVA_VLA Vulnerability analysis

AVA_VLA.4 Highly resistant

AVA_VLA.4.1D The developer shall perform a vulnerability analysis.

AVA_VLA.4.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.4.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.4.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.4.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.4.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

SIMEOS SECURITY TARGET

AVA_VLA.4.5C The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.

AVA_VLA.4.6C The vulnerability analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

AVA_VLA.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.4.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.4.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a high attack potential.

5.3 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

5.3.1 IT environment functional requirements

5.3.1.1 Java Card

5.3.1.1.1 Bytecode verification (BCVG)

FDP_IFC.2/BCV Complete information flow control

FDP_IFC.2.1/BCV The TSF shall enforce the **TYPING information flow control SFP** on **S.LOCVAR, S.STCKPOS, S.FLD, S.MTHD** and all operations that cause that information to flow to and from subjects covered by the SFP.

Non editorial refinement:

Subjects (prefixed with an "S") covered by this policy are:

Subject	Description
S.LOCVAR	Any local variable of the currently executed method.
S.STCKPOS	Any operand stack position of the currently executed method.
S.FLD	Any field declared in a package loaded on the card.
S.MTHD	Any method declared in a package loaded on the card.

The operations (prefixed with " OP ") that make information flow between the subjects are all bytecodes. For instance, the *aload_0* bytecode causes information to flow from the local variable 0 to the top of the operand stack; the bytecode *putfield(x)* makes information flow from the top of the operand stack to the field x; and the *return_a* bytecode makes information flow out of the currently executed method.

SIMEOS SECURITY TARGET

Operation	Description
<i>OP.BYTECODE(BYTCD)</i>	Any bytecode for the Java Card platform ("Java Card bytecode").

The information (prefixed with an " / ") controlled by the typing policy are the bytes, shorts, integers, references and return addresses contained in the different storage units of the JCVM (local variables, operand stack, static fields, instance fields and array positions).

Information	Description
<i>I.BYTE(BY)</i>	Any piece of information that can be encoded in a byte.
<i>I.SHORT(SH)</i>	Any piece of information that can be encoded in a short value.
<i>I.INT(W1,W2)</i>	Any piece of information that can be encoded in an integer value, which in turn is encoded in two words w1 and w2.
<i>I.REFERENCE(RF)</i>	Any reference to a class instance or an array.
<i>I.ADDRESS(ADRS)</i>	Any return address of a subroutine.

FDP_IFC.2.2/BCV The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

FDP_IFF.2/BCV Hierarchical security attributes

FDP_IFF.2.1/BCV The TSF shall enforce the **TYPING information flow control SFP** based on the following types of subject and information security attributes: **(1) type attribute of the information, (2) type attribute of the storage units of the JCVM, (3) class attribute of the fields and methods, (4) bounds attribute of the methods.**

Non editorial refinement:

The following table describes which security attributes are attached to which subject/information of our policy.

Subject / Information	Attributes
<i>S.LOCVAR</i>	<i>TYPE</i>
<i>S.STCKPOS</i>	<i>TYPE</i>
<i>S.FLD</i>	<i>TYPE, CLASS</i>
<i>S.MTHD</i>	<i>TYPE, CLASS, BOUNDS</i>
<i>I.BYTE(BY)</i>	<i>TYPE</i>
<i>I.SHORT(SH)</i>	<i>TYPE</i>
<i>I.INT(W1,W2)</i>	<i>TYPE</i>
<i>I.REFERENCE(RF)</i>	<i>TYPE</i>
<i>I.ADDRESS(ADRS)</i>	<i>TYPE</i>

The following table describes the security attributes.

SIMEOS SECURITY TARGET

Attribute Name	Description
<i>TYPE</i>	Either the type attached to the information, or the type held or declared by the subject.
<i>CLASS</i>	The class where a field or method is declared.
<i>BOUNDS</i>	The start and end of the method code inside the method component of the CAP file where it is declared.

The *TYPE* security attribute attached to local variables and operand stack positions is the type of information they currently hold. The *TYPE* attribute of the fields and the methods is the type declared for them by the programmer.

The *BOUNDS* attribute of a method is used to prevent control flow to jump outside the currently executed method.

The following table describes the possible values for each security attribute.

Name	Description
<i>TYPE</i>	byte, short, int ₁ , int ₂ , any class name <i>C</i> , <i>T</i> [] with <i>T</i> any type in the Java Card platform ("Java Card type"), T ₀ (T ₁ x ₁ ,... T _n x _n) with T ₀ ,... T _n any Java Card type, RetAddr(<i>adrs</i>), Top, Null, ⊥.
<i>CLASS</i>	The name of a class, represented as a reference into the class Component of one of the packages loaded on the card.
<i>BOUNDS</i>	Two integers marking a rank into the method component of a package loaded on the card.

Byte values have type **byte** and short values have type **short**. The first and second halves of an integer value has respectively type **int₁**, and **int₂**. The type of a reference to an instance of the class *C* is **C** itself. A reference to an array of elements of type *T* has type **T[]**. From the previous basic types it is possible to build the type T₀ (T₁ x₁,... T_n x_n) of a method. A return address *adrs* of a subroutine has type **RetAddr(*adrs*)**. Finally, the former Java Card types are extended with three extra types **Top**, **Null** and **⊥**, so that the domain of types forms a complete lattice. **Top** is the type of any piece of data, that is, the maximum of the lattice. **Null** is the type of the default value null of all the reference types (classes and arrays). **⊥** is the type of an element that belongs to all types (for instance the value 0, provided that null is represented as zero).

FDP_IFF.2.2/BCV The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

The following rules constitute a synthetic formulation of the information flow control:

R.JAVA.6 If the bytecode pushes values from the operand stack, then there are a sufficient number of values on the stack and the values of the attribute *TYPE* of the top positions of the stack is appropriate with respect to the ones expected by the bytecode.

R.JAVA.7 If the bytecode pushes values onto the operand stack, then there is sufficient room on the operand stack for the new values. The values, with the appropriate attribute *TYPE* value are added to the top of the operand stack.

R.JAVA.8 If the bytecode modifies a local variable with a value with attribute *TYPE* *T*, it must be recorded that the local variable now contains a value of that type. In addition, the variable shall be among the local variables of the method.

R.JAVA.9 If the bytecode reads a local variable, it must be ensured that the specified local variable contains a value with the attribute *TYPE* specified by the bytecode.

R.JAVA.10 If the bytecode uses a field, it must be ensured that its value is of an appropriate type. This type is indicated by the *CLASS* attribute of the field.

R.JAVA.11 If the bytecode modifies a field, then it must be ensured that the value to be assigned is of an appropriate type. This type is indicated by the *CLASS* attribute of the field

R.JAVA.12 If the bytecode is a method invocation, it must be ensured that it is invoked with arguments of the appropriate type. These types are indicated by the *TYPE* and *CLASS* attributes of the method.

SIMEOS SECURITY TARGET

R.JAVA.13 If the bytecode is a branching instruction, then the bytecode target must be defined within the *BOUNDS* of the method in which the branching instruction is defined.

FDP_IFF.2.3/BCV The TSF shall enforce the (following additional information flow control SFP rules): **none**.

FDP_IFF.2.4/BCV The TSF shall provide the following (list of additional SFP capabilities): **none**.

FDP_IFF.2.5/BCV The TSF shall explicitly authorise an information flow based on the following rules: **none**.

FDP_IFF.2.6/BCV The TSF shall explicitly deny an information flow based on the following rules: **none**.

FDP_IFF.2.7/BCV The TSF shall enforce the following relationships for any two valid information flow control security attributes:

- a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and
- b) There exists a "least upper bound" in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and
- c) There exists a "greatest lower bound" in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

Application note:

FDP_IFF.2.2

The rules described above are strongly inspired in the rules described in section 4.9 of [JVM], Second Edition. The complete set of typing rules can be derived from the "Must" clauses from Chapter 7 of [JCVM21] as instances of the rules defined above.

FDP_IFF.2.7

The order relationship between Java Card types is described, for instance, in the description of the **checkcast** bytecode of [JCVM22]. That relation is with the following rules:

- **Top** is the maximum of all types;
- **Null** is the minimum of all classes and array types;
- \perp is the minimum of all types.

These three extra types are introduced in order to satisfy the two last items in requirement FDP_IFF.2.7.

FMT_MSA.1/BCV.1 Management of security attributes
--

FMT_MSA.1.1/BCV.1 The TSF shall enforce the **TYPING information flow control SFP** to restrict the ability to **modify** the security attributes **TYPE security attribute of the fields and methods** to **none**.

SIMEOS SECURITY TARGET

FMT_MSA.1/BCV.2 Management of security attributes

FMT_MSA.1.1/BCV.2 The TSF shall enforce the **TYPING information flow control SFP** to restrict the ability to **modify** the security attributes **TYPE security attribute of local variables and operand stack position to the role Bytecode Verifier**.

Application note:

1. See *FMT_SMR.1.1* for the roles.

2. *FMT_MSA.1.1/BCV.2*. The TYPE attribute of the local variables and the operand stack positions is identified to the attribute of the information they hold. Therefore, this security attribute is possibly modified as information flows. For instance, the rules of the typing function enable information to flow from a local variable *lv* to the operand stack by the operation *sload*, provided that the value of the type attribute of *lv* is **short**. This operation hence modifies the type attribute of the top of the stack. The modification of the security attributes should be done according to the typing rules derived from Chapter 7 of [JCVM21].

FMT_MSA.2/BCV Secure security attributes

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

Application note:

During the type verification of a method, the bytecode verifier makes intensive use of the information provided in the CAP format like the sub-class relationship between the classes declared in the package, the type and class declared for each method and field, the rank of exceptions associated to each method, and so on. All that information can be thought of as security attributes used by the bytecode verifier, or as information relating security attributes. Moreover, the bytecode verifier relies on several properties about the CAP format. All the properties on the CAP format required by the bytecode verifier could, for instance, be completely described before starting type verifications. Examples of such properties are:

- Correspondences between the different components of the CAP file (for instance, each class in the class component has an entry in the descriptor component).
- Pointer soundness (example: the index argument in a static method invocation always has an entry in the constant pool);
- Absence of hanged pointers (example: each exception handler points to the beginning of some bytecode);
- Redundant information (enabling different ways of searching for it);
- Conformance to the Java Language Specification respecting the access control features mentioned in §2.2 of [JCVM22].
- Packages that are loaded post-issuance can not contain native code.

FMT_MSA.3/BCV Static attribute initialisation

FMT_MSA.3.1/BCV The TSF shall enforce the **TYPING information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/BCV The TSF shall allow the **none** to specify alternative initial values to override the default values when an object or information is created.

Application note:

SIMEOS SECURITY TARGET

FMT_MSA.3.1 The TYPE attribute of the fields and methods is fixed by the application provider and never modified. When a method is invoked, the operand (type) stack is empty. The initial type assigned to those local variables that correspond to the method parameters is the type the application provider declared for those parameters. Any other local variable used in the method is set to the default value Top.

FMT_MSA.3.2 The intent is to have none of the identified roles to have privileges with regards to the default values of the TYPE attributes.

FMT_SMR.1/BCV Security roles

FMT_SMR.1.1/BCV The TSF shall maintain the roles **Bytecode Verifier**.

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

FRU_RSA.1/BCV Maximum quotas

FRU_RSA.1.1/BCV The TSF shall enforce maximum quotas of the following resources: **the operand stack and the local variables** that **individual user** can use **simultaneously**.

Non editorial refinement:

Individual user is a **method** here.

SIMEOS SECURITY TARGET

6. TOE SUMMARY SPECIFICATION

6.1 TOE SECURITY FUNCTIONS

The security functions of the product are those of both the hardware and the software platforms. The Security Functions that reply to the SFR of the IC are described in [ST/Infineon].

The minimum strength for the security functions is SOF-high.

6.1.1 Card Operation

This section contains security functions dedicated to the operating system that may be used by all the applications on the card.

SF.SmartCardPlatform

The TOE runs tests at power on to check it has not been corrupted.

The TOE provides the ability to check the integrity of sensitive data stored in the card.

The TOE hides sensitive data transfers and operations from outside.

The TOE is protected against power and electromagnetic analysis, DFA & timing attacks.

It detects physical tampering and provides automatic response.

The data or operations protection against attacks uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

SF.CardManager

The Card Manager controls the flow of APDUs between off card applications and the current application (native or JC applet) and the access to the APDU buffer.

The Card Manager controls the card and applets Life Cycles.

The Card Manager loads applets into the card on behalf of the Byte Code Verifier. The Card Manager locks the loading of applets on the card on behalf of the Issuer.

The Card Manager loads OP keys into the card on behalf of the Operator.

This SF uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

Application note:

Usimera Protect doesn't support the extradition mechanism.

SF.TrustedChannel

This SF establishes a trusted channel between the card and a remote IT product. The trusted channel is [GP] compliant. It protects the integrity and/or the confidentiality of data loading into the card.

Additionally, a DAP mechanism protects the integrity of package loading and the Non repudiation of origin.

Keys are protected in Integrity and confidentiality when loaded.

The GP secure channel uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

SIMEOS SECURITY TARGET

6.1.2 Common services

Security functions serving both the OS and the JCS.

SF.Transaction

This SF manages transactions in the:

- o OS;
- o JCVM. It enables to create Java Objects within a transaction. Transaction management includes the rollback of operations according to [JCRE221].

The TSF preserves a secure state when failures occur.

This SF does not use probabilistic or permutational effects.

This function has no strength.

6.1.3 Mob Com

SF.Authentication

This Security Function enforces the authentication of the:

- o SIM application to the network, using Comp128V2, Comp128V3 or GSM-Milenage algorithms;
- o USIM application to the network, using Milenage algorithm.

This SF is invoked by the RunGSMAlgorithm GSM APDU command for a 2G network and by the Authenticate 3G APDU command for a 3G network respectively.

- o 2G. As defined in [TS 51.011], it computes two bytes sequences called S_{RES} and Kc from the random bytes sequences provided as the APDU data and the authentication key associated to the SF. The command is blocked and the key is no longer available when the authentication counter attached to the key reaches its maximum value.
- o 3G. As defined in [TS 31.102], it computes three bytes sequences called RES, Ck and Ik from the random bytes sequences provided as the APDU data and the authentication key associated to the SF. The command is blocked and the key is no longer available when the MAC counter attached to the key reaches its maximum value.

This function admits two sub-functions:

- o sSF.Crypto_Operation, activated by a cryptographic request;
- o sSF.Material_Association, ensuring the two ways relationship between an user account related application and the associated key.

The cryptographic operations of 2G or 3G network authentication use probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

6.1.4 Java Card

SF.Attributes

This SF manages the values of the following security attributes: resident applets, active applets and currently selected applet. It also checks the consistency of applets' life cycle.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.Crypto

This SF provides the crypto functions and their Java API. This includes:

- o Key generation, Import, Export, and Deletion;
- o Cryptographic operations.

SIMEOS SECURITY TARGET

for the following algorithms: TDES 2 keys, AES 128 and SHA-1.

This SF uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

Application note:

The TOE executes also DES (one key), MD5 and CRC, but these algorithms are not strong enough for SOF high and therefore no FCS_COP functional requirement was created for any of them. These algorithms are considered as services and their usage requires special care (see AGD class).

SF.Erase

This SF ensures that sensitive data cannot be accessed upon and after some types of operations. This SF may be split in:

- o Logical deletion upon package and/or applet(s) deletion,
- o Physical deletion upon and after object(s) deletion.

This SF erases the APDU buffer and the cryptographic buffer.

The hiding operations use probabilistic or permutational effects and have to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

SF.Firewall

The JCRE firewall enforces applet isolation. The JCRE shall allocate and manage a context for each *applet* or *package* installed respectively loaded on the card and its own JCRE context. *Applets* cannot access each other's objects unless they are defined in the same package (they share the same context) or they use the object sharing mechanism supported by JCRE.

This SF participates to information confidentiality.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.Install

This SF ensures that the package loading is performed safely without loss, substitution, addition, modification, repetition of data or any other integrity failure on the loaded data such as a wrong order in the delivery of data by incoming APDUs.

It also ensures a safe *applet* installation process.

It modifies the CAP files in a safe way and performs coherency checks on the CAP files. It verifies the export references of the packages upon linking them.

This SF ensures that Java Card objects such as classes, *packages* and *applet* instances use limited resources.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.JCRE

The JCRE owns JCRE entry points objects of which methods may be accessed by any context. Their fields may only be accessed by the JCRE context.

The reference of temporary JCRE entry point objects and global arrays cannot be stored in class variables, instance fields or array components.

The JCRE is the only one allowed to create JCRE entry point objects and global arrays and to define them as temporary or permanent.

The JCRE has access to any objects and methods owned by any context. But it only invokes the methods `select()`, `process()`, `deselect()`, `getShareableInterfaceObject()` defined in `Applet` class and `MultiSelectable` interface.

The JCRE is the currently active context when the JCVN starts running after a card reset.

SIMEOS SECURITY TARGET

It is the only context allowed to register applets.

The *JCRE* supplies transient memory management through JC System services.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.Applet

The SF defines the behaviour of each Java Card application through a strictly defined interface named Applet class and to the management of the *AIDs*. Each new application inherits its behaviour and the associated constraints from the Applet class model.

This SF realises applet code interpretation. From the interpretation point of view, the *applet*'s code is considered as data to read. There is no way for an *applet* to be executed independently or to access platform resources. Thus the *JCRE* decides to start and to stop the applet execution.

This SF participates to information confidentiality, information integrity and availability.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.Domain

This SF consists in memory management with the *JCRE* scope. It is realized thanks to OS services.

The *applet* needs some different types of storage areas to be executed. As *applet* has no direct access to internal resources in particular to memory, the *JCRE* manages the memory for the applets. For this purpose, some services have been defined. On its side, the *JCRE* has its own execution space in memory that is not accessible to the *applets*.

This SF does not use probabilistic or permutational effects.

This function has no strength.

SF.PIN

This SF supplies to *applet* a mean to assume a user identification and authentication with the OwnerPin class.

It is supplied through a secure comparison between a PIN stored in the persistent memory and a data received from the *CAD*.

This SF participates to information confidentiality.

This SF uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF high.

The strength of this function is SOF-high.

SF.JCRMI

This SF handles RMI security features: access control, information flow control, the related security attributes and their management.

This SF does not use probabilistic or permutational effects.

This function has no strength.

6.2 ASSURANCE MEASURES

6.2.1 ASE

ST_SIMEOS

SIMEOS Security Target

SIMEOS SECURITY TARGET

6.2.2 ACM**AUT_SIMEOS**

SIMEOS Partial CM automation

CAP_SIMEOS

SIMEOS Generation support and acceptance procedures

SCP_SIMEOS

SIMEOS Problem tracking CM coverage

6.2.3 ADO**DEL_SIMEOS**

SIMEOS Detection of modification (Procedures for delivery)

IGS_SIMEOS

SIMEOS Installation, Generation and Start Up Procedures

6.2.4 ADV**FSP_SIMEOS**

SIMEOS Functional Specification

HLD_SIMEOS

SIMEOS High-level Design

LLD_SIMEOS

SIMEOS Low-level Design

IMP_SIMEOS

SIMEOS Implementation representation

SPM_SIMEOS

SIMEOS Security Policy Model

6.2.5 AGD**USR_SIMEOS**

SIMEOS User Guidance

ADM_SIMEOS

SIMEOS Administrator Guidance

SIMEOS SECURITY TARGET

6.2.6 ALC**DVS_SIMEOS**

SIMEOS Identification of security measures (Development Security Documentation)

LCD_SIMEOS

SIMEOS Developer defined life-cycle model (Life-cycle definition documentation)

TAT_SIMEOS

SIMEOS Well-defined development tools (Documentation of development tools)

6.2.7 ATE**COV_SIMEOS**

SIMEOS Analysis of test coverage

DPT_SIMEOS

SIMEOS Analysis of the depth of testing

FUN_SIMEOS

SIMEOS Test Documentation

IND_SIMEOS

Independent testing - sample

6.2.8 AVA**MSU_SIMEOS**

SIMEOS Analysis and testing for insecure states

SOF_SIMEOS

SIMEOS Strength of TOE security functions analysis

VLA_SIMEOS

SIMEOS Vulnerability Analysis

SIMEOS SECURITY TARGET

7. PP CLAIMS

[PP/JCS] Java Card™ System, Protection Profile Collection, Version 1.0b

7.1 PP REFERENCE

[PP/JCS] is claimed. The Java Card TOE defined into this PP is extended to the actual TOE; consequently it covers the operating system and the mobile communication applications.

7.2 PP ADDITIONS

Additional Security Functional Requirements

The following SFR have been added to the ST main groups of SFRs compared to [PP/JCS]:

Java Card part:

- *FMT_SMF.1/CM*. It is required as dependency by *FMT_MSA.1*.

Mobile Communications part:

- *FCS_COP.1/COMP128V2*
- *FCS_COP.1/COMP128V3*
- *FCS_COP.1/MILENAGE2G*
- *FCS_COP.1/MILENAGE*
- *FDP_ACC.1/Auth*
- *FDP_ACF.1/Auth*
- *FDP_SDI.2/Auth*
- *FIA_UID.1/Auth*
- *FMT_MSA.1/Auth*
- *FMT_MSA.2/Auth*
- *FMT_MSA.3/Auth*
- *FMT_SMR.1/Auth*
- *FPR_UNO.1/Auth*
- *FPT_FLS.1/Auth*
- *FPT_ITT.1/Auth*
- *FPT_ITT.3/Auth*

Operations that have been completed on SFRs from [PP/JCS]:

CoreG:

- Firewall Policy: *FDP_IFF.1/JCVM*;
- Application Programming Interface: *FCS_CKM.1*, *FCS_CKM.2*, *FCS_CKM.3*, *FCS_CKM.4* and *FCS_COP.1* have been instantiated with the algorithms available on the card.
- Card Security Management: *FDP_SDI.2*, *FPR_UNO.1*.

RMI (RMIG):

- JCRMI Policy: *FDP_IFF.1/JCRMI*

Secure carrier (CarG):

- *FCO_NRO.2/CM*, *FDP_IFF.1/CM*, *FMT_MSA.1/CM*.

SIMEOS SECURITY TARGET

Smart card platform (SCPG):

- *FPT_FLS.1/SCP, FRU_FLT.1/SCP, FPT_PHP.3/SCP.*

Card manager (CMGRG):

- *FDP_ACF.1/CMGR, FMT_MSA.1/CMGR, FMT_MSA.3/CMGR.*

The following SFRs have been merged and refined when necessary, with respect to [PP/JCS]: Please see the group *Merged SFRs* from the Java Card part.

Additional Assurance requirements

[PP/JCS] requires EAL4 augmented with:

- ADV_IMP.2
- AVA_VLA.3.

Additionally, this ST requires:

- ALC_DVS.2
- ADV_IMP.2
- AVA_MSU.3
- AVA_VLA.4

SIMEOS SECURITY TARGET

8. RATIONALE

8.1 SECURITY OBJECTIVES RATIONALE

Not delivered in public version.

8.2 SECURITY REQUIREMENTS RATIONALE

Not delivered in public version.

8.3 TOE SUMMARY SPECIFICATION RATIONALE

Not delivered in public version.

SIMEOS SECURITY TARGET

Notice

This document has been generated with TL SET version 1.7.1, CC version 2.2 with interpretations of January 2004 (including interpretations: 137). www.trusted-logic.com.

SIMEOS SECURITY TARGET

Index

A

A.APPLET 22
 A.NET-OP_KEY_MGT 22
 A.VERIFICATION 22
 ACM_AUT.1 61
 ACM_CAP.4 62
 ACM_SCP.2 63
 ADM_SIMEOS 84
 ADO_DEL.2 63
 ADO_IGS.1 63
 ADV_FSP.2 64
 ADV_HLD.2 64
 ADV_IMP.2 65
 ADV_LLD.1 66
 ADV_RCR.1 66
 ADV_SPM.1 67
 AGD_ADM.1 67
 AGD_USR.1 68
 ALC_DVS.2 69
 ALC_LCD.1 69
 ALC_TAT.1 69
 ATE_COV.2 70
 ATE_DPT.1 70
 ATE_FUN.1 71
 ATE_IND.2 71
 AUT_SIMEOS 83
 AVA_MSU.3 72
 AVA_SOF.1 72
 AVA_VLA.4 73

C

CAP_SIMEOS 84
 COV_SIMEOS 85

D

D.API_DATA 20
 D.APP_C_DATA 19
 D.APP_CODE 19
 D.APP_I_DATA 19
 D.APP_KEYS 19
 D.AUTH_CODE 20

D.AUTH-APPLI_ATTRIB 20
 D.AUTH-CRYPTO_ALGO 20
 D.AUTH-KEY_ATTRIB 20
 D.CRYPTO 21
 D.JCS_CODE 20
 D.JCS_DATA 20
 D.JCS_KEYS 20
 D.NET_ACCESS 19
 D.NET_AUTH_KEY 19
 D.PIN 19
 D.SEC_DATA 20
 DEL_SIMEOS 84
 DPT_SIMEOS 85
 DVS_SIMEOS 84

F

FAU_ARP.1/JCS 43
 FCO_NRO.2/CM 58
 FCS_CKM.1/AES 40
 FCS_CKM.1/TDES 41
 FCS_CKM.2/AES 41
 FCS_CKM.2/TDES 41
 FCS_CKM.3/AES 41
 FCS_CKM.3/TDES 41
 FCS_CKM.4/AES 41
 FCS_CKM.4/TDES 41
 FCS_COP.1/AES 42
 FCS_COP.1/COMP128V2 33
 FCS_COP.1/COMP128V3 33
 FCS_COP.1/MILENAGE 34
 FCS_COP.1/MILENAGE2G 33
 FCS_COP.1/SHA-1 42
 FCS_COP.1/TDES 42
 FDP_ACC.1/Auth 34
 FDP_ACC.1/CMGR 32
 FDP_ACC.2/ADEL 47
 FDP_ACC.2/FIREWALL 54
 FDP_ACC.2/JCRMI 50
 FDP_ACF.1/ADEL 48
 FDP_ACF.1/Auth 34
 FDP_ACF.1/CMGR 32

SIMEOS SECURITY TARGET

FDP_ACF.1/FIREWALL 55	FMT_MSA.3/BCV 78
FDP_ACF.1/JCRMI 51	FMT_MSA.3/CM 60
FDP_IFC.1/JCRMI 52	FMT_MSA.3/CMGR 33
FDP_IFC.1/JCVM 39	FMT_MSA.3/FIREWALL 40
FDP_IFC.2/BCV 74	FMT_MSA.3/JCRMI 53
FDP_IFC.2/CM 58	FMT_MTD.1/JCRE 45
FDP_IFF.1/CM 59	FMT_MTD.3 45
FDP_IFF.1/JCRMI 52	FMT_REV.1/JCRMI 54
FDP_IFF.1/JCVM 39	FMT_SMF.1/CM 37
FDP_IFF.2/BCV 75	FMT_SMR.1/Auth 36
FDP_ITC.2/Installer 46	FMT_SMR.1/BCV 78
FDP_RIP.1/ABORT 42	FMT_SMR.1/JCS 38
FDP_RIP.1/ADEL 50	FPR_UNO.1 44
FDP_RIP.1/APDU 42	FPR_UNO.1/Auth 36
FDP_RIP.1/bArray 43	FPT_AMT.1/SCP 31
FDP_RIP.1/KEYS 43	FPT_FLS.1/ADEL 50
FDP_RIP.1/OBJECTS 40	FPT_FLS.1/Auth 36
FDP_RIP.1/ODEL 57	FPT_FLS.1/Installer 47
FDP_RIP.1/TRANSIENT 57	FPT_FLS.1/JCS 44
FDP_ROL.1/FIREWALL 43	FPT_FLS.1/ODEL 58
FDP_SDI.2 44	FPT_FLS.1/SCP 31
FDP_SDI.2/Auth 35	FPT_ITT.1/Auth 36
FDP_UIT.1/CM 60	FPT_ITT.3/Auth 37
FIA_ATD.1/AID 45	FPT_PHP.3/SCP 31
FIA_UID.1/Auth 35	FPT_RCV.3/JCS 38
FIA_UID.1/JCS 37	FPT_RCV.4/SCP 32
FIA_UID.2/AID 45	FPT_RVM.1/SCP 32
FIA_USB.1 46	FPT_SEP.1/SCP 31
FMT_MSA.1/ADEL 49	FPT_TDC.1 44
FMT_MSA.1/Auth 35	FPT_TST.1 45
FMT_MSA.1/BCV.1 77	FRU_FLT.1/SCP 31
FMT_MSA.1/BCV.2 77	FRU_RSA.1/BCV 78
FMT_MSA.1/CM 60	FRU_RSA.1/Installer 47
FMT_MSA.1/CMGR 33	FSP_SIMEOS 84
FMT_MSA.1/EXPORT 53	FTP_ITC.1/CM 61
FMT_MSA.1/JCRE 57	FUN_SIMEOS 85
FMT_MSA.1/JCRMI 53	H
FMT_MSA.1/REM_REFS 53	HLD_SIMEOS 84
FMT_MSA.2/Auth 35	I
FMT_MSA.2/BCV 77	IGS_SIMEOS 84
FMT_MSA.2/JCRE 40	IMP_SIMEOS 84
FMT_MSA.3/ADEL 49	IND_SIMEOS 85
FMT_MSA.3/Auth 35	

SIMEOS SECURITY TARGET

L

LCD_SIMEOS 84

LLD_SIMEOS 84

M

MSU_SIMEOS 85

O

OE.APPLET 29

OE.DEVELOPMENT 29

OE.KEY_MANAGEMENT 29

OE.VERIFICATION 30

OSP.KeyMgt_DEV_MobCom 25

OSP.NATIVE 25

OSP.VERIFICATION 25

OT.ACCOUNT_NET_ACCESS 26

OT.ALARM 27

OT.AUTH-ALGO_CONFID 26

OT.AUTH-ALGO_EXEC 26

OT.AUTH-ALGO_INTEG 26

OT.AUTH-KEYS_CONFID 26

OT.AUTH-KEYS_INTEG 26

OT.CARD-MANAGEMENT 29

OT.CIPHER 27

OT.DELETION 28

OT.FIREWALL 27

OT.INSTALL 28

OT.KEY-MNGT 28

OT.LOAD 28

OT.NATIVE 27

OT.NATIVE.2 29

OT.OBJ-DELETION 28

OT.OPERATE 26

OT.PIN-MNGT 28

OT.REALLOCATION 27

OT.REMOTE 28

OT.RESOURCES 26

OT.SCP.IC 29

OT.SCP.RECOVERY 29

OT.SCP.SUPPORT 29

OT.SHRD_VAR_CONFID 27

OT.SHRD_VAR_INTEG 27

OT.SID 26

OT.TRANSACTION 27

S

S.ADEL 21

S.BCV 21

S.CAD 21

S.CRD 21

S.JCRE 21

S.Operator 21

S.PACKAGE 21

S.SPY 22

S.User_Account 21

SCP_SIMEOS 84

SF.Applet 83

SF.Attributes 81

SF.Authentication 81

SF.CardManager 80

SF.Crypto 81

SF.Domain 83

SF.Erase 82

SF.Firewall 82

SF.Install 82

SF.JCRE 82

SF.JCRMI 83

SF.PIN 83

SF.SmartCardPlatform 80

SF.Transaction 81

SF.TrustedChannel 80

SOF_SIMEOS 85

SPM_SIMEOS 84

ST_SIMEOS 83

T

T.AUTH-ALGO_DISCLOSURE 23

T.AUTH-ALGO_EXE 23

T.AUTH-ALGO_MODIFICATION 23

T.AUTH-DATA_DISCLOSURE 23

T.AUTH-DATA_MODIFICATION 23

T.CONFID-APPLI-DATA 23

T.CONFID-JCS-CODE 23

T.CONFID-JCS-DATA 23

T.DELETION 25

T.EXE-CODE.1 24

T.EXE-CODE.2 24

SIMEOS SECURITY TARGET

T.EXE-CODE-REMOTE 24

T.INSTALL 25

T.INTEG-APPLI-CODE 24

T.INTEG-APPLI-CODE.2 24

T.INTEG-APPLI-DATA 24

T.INTEG-APPLI-DATA.2 24

T.INTEG-JCS-CODE 24

T.INTEG-JCS-DATA 24

T.NATIVE 24

T.OBJ-DELETION 25

T.PHYSICAL 22

T.RESOURCES 24

T.SID.1 24

T.SID.2 24

T.UNAUTH_NET_ACCESS 23

TAT_SIMEOS 85

U

USR_SIMEOS 84

V

VLA_SIMEOS 85

END OF SECURITY TARGET