
# UpTeq NFC3.2.2_Generic v1.0

# USIM Platform

## Common Criteria / ISO 15408

## Security Target – Public version

## EAL4+

Copyright Gemalto SA – 2016

# TABLE OF CONTENTS

# TABLE OF FIGURES



# TABLE OF TABLES

# 1. Reference documents

## 1.1. EXTERNAL REFERENCES [ER]

| [ISO] | ISO references |
|---|---|
| [ISO7816] | Identification cards – Integrated circuit(s) cards with contacts - Books 1 to 9 |
| [ISO14443] | Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Books 1 to 4 |
| **[Javacard]** | **Javacard references** |
| [JCRE301] | Runtime Environment Specification<br>Java Card Platform, Version 3.0.1, Classic Edition, May 2009 |
| [JCVM301] | Virtual Machine Specification<br>Java Card Platform, Version 3.0.1, Classic Edition, May 2009 |
| [JCAPI301] | Java Card APIs specification,<br>Version 3.0.1, Classic Edition, May 2009 |
| **[GP]** | **Global Platform references** |
| [GP_22] | GlobalPlatform Card Specification<br>Version 2.2.1, January 2011 |
| [GP22_AmdA] | Confidential Card Content Management - GlobalPlatform Card Specification v2.2 – Amendment A<br>Version 1.0.1, January 2011 |
| [GP22_AmdB] | Card Remote Application Management over HTTP – GlobalPlatform Card Specification v2.2 – Amendment B<br>Version 1.1.2, May 2014 |
| [GP22_AmdC] | Card Contactless Services Card Specification v2.2 – Amendment C<br>Version 1.1.1, July 2014 |
| [GP22_AmdD] | Secure Channel Protocol 03 – Global Platform Card Specification v2.2 – Amendment D<br>Version 1.1, September 2009 |
| [GP22_AmdE] | Security Upgrade for Card Content Management – Global Platform Card Specification v2.2 – Amendment E<br>Version 1.0.1, July 2014 |
| [GP22_UICC] | GlobalPlatform UICC Configuration<br>Version 1.0.1, January 2011 |
| **[ETSI]** | **ETSI references** |
| [TS 101.220] | ETSI numbering system for telecommunication application providers<br>Version 8.3.0 |
| [TS 102.124] | Transport Protocol for UICC based Applications; Stage 1<br>Version 7.1.0 |
| [TS 102.127] | Transport protocol for CAT applications; Stage 2<br>Version 6.10.0 |
| [TS 102.221] | UICC-Terminal interface; Physical and logical characteristics<br>Version 7.5.0, Version 7.10.0 (Partial) |
| [TS 102.222] | Integrated Circuit Cards (ICC); Administrative commands for telecommunications applications<br>Version 6.11.0 |
| [TS 102.223] | Application Toolkit (CAT)<br>Version 8.1.0, Version 8.3.0 (Partial), Version 12.0.0 (Partial) |
| [TS 102.224] | Security mechanisms for UICC based Applications - Functional requirements<br>Version 7.1.0 |
| [TS 102.225] | Secured packet structure for UICC based applications<br>Version 12.0.0 |
| [TS 102.226] | Remote APDU structure for UICC based applications<br>Version 11.2.0 (Partial) |
| [TS 102.230] | UICC-Terminal interface: Physical, electrical and logical test specification<br>Version 5.9.0 |

| | |
|---|---|
| [TS 102.240] | UICC Application Programming Interface and Loader Requirements; Service description<br>Version 7.0.0 + Release 8 (Partial) |
| [TS 102.241] | UICC Application Programming Interface (UICC API) for Java Card<br>Version 9.2.0 |
| [TS 102.267] | Connection oriented service API (COS)<br>Version 7.1.0 |
| [TS 102.310] | Extensible Authentication Protocol support in the UICC<br>Version 7.0.0 |
| [TS 102.588] | Application invocation Application Programming Interface (API) by a UICC web server for Java Card™ platform;<br>Version 7.2.0 |
| [TS 102.613] | SWP Single Wire Protocol<br>Version 10.0.0 |
| [TS 102.622] | HCI Host Controller Interface Architecture<br>Version 12.0.0 (Partial) |
| [TS 102.705] | Contactless API for the "Java Card(TM) Platform"<br>Version 11.0.0 |
| **[3GPP]** | **3GPP references** |
| [TS 21.111] | Technical Specification Group Terminals; USIM and IC card requirements<br>Version 7.1.0 |
| [TS 23.048] | Security mechanisms for the (U)SIM application toolkit; Stage 2<br>Version V5.9.0 |
| [TS 31.101] | UICC-terminal interface; Physical and logical characteristics<br>Version 9.1.0 |
| [TS 31.102] | Characteristics of the USIM application<br>Version 9.6.0 (Partial) |
| [TS 31.103] | Characteristics of the IP Multimedia Services Identity Module (ISIM) application<br>Version 8.0.1 (Partial) |
| [TS 31.111] | Universal Subscriber Identity Module Application Toolkit (USAT)<br>Version 12.3.0 (Partial) |
| [TS 31.115] | Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications<br>Version 12.0.0 (Partial) |
| [TS 31.116] | Remote APDU Structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications<br>Version 11.1.0 (Partial) |
| [TS 31.130] | (U)SIM Application Programming Interface API; (U)SIM API for Java Card(TM)<br>Version 10.4.0 (Partial) |
| [TS 31.133] | IP Multimedia Services Identity Module (ISIM) API<br>Version 9.2.0 |
| [TS 31.900] | SIM/USIM internal and external interworking aspects<br>Version 8.0.0 |
| [TS 31.919] | 2G/3G Java Card(TM) Application Programming Interface (API) based applet interworking<br>Version 8.0.0 |
| [TS 33.102] | 3G security; Security architecture<br>Version 8.2.0 |
| [TS 33.103] | 3G security; Integration guidelines<br>Version 4.2.0 (Release 5) |
| [TS 33.105] | Cryptographic algorithm requirements<br>Version 8.0.0 |
| [TS 43.019] | SIM API For JavaCard<br>Version 5.6.0 |
| **[MIFARE]** | **MIFARE DESFire EV1 / M4M references** |
| [DESFire_IS] | MIFARE DESFire EV1 – Interface Specification<br>Rev 1.0, 21.11.2008 |

| | |
|---|---|
| [MIFARE_VC-CREAT] | MIFARE Virtual Card Creation<br>Rev 1.0.2, February 11th 2014 |
| [MIFARE_VC-MNGT] | MIFARE Virtual Card Management Host API – Functional Specification<br>Rev 1.0.2 |
| [DESFire_HIS] | MIFARE DESFire EV1 Host Interface – Functional specification<br>Rev 1.0.1 |
| [MIFARE_JCAPI] | MIFARE Java Card API, MIFARE4Mobile inter industry group |
| [M4M_ARC] | MIFARE4Mobile architecture<br>Rev 2.1.1 |
| [M4M_REM-MNGT] | MIFARE4Mobile Remote Management API<br>Rev 2.1.1 |
| [M4M-WALLET] | MIFARE4Mobile Wallet API<br>Rev 2.1.1 |
| **[CC]** | **Common Criteria references** |
| [CC-1] | Common Criteria for Information Technology Security Evaluation<br>Part 1: Introduction and general model, CCMB-2012-09-001, Version 3.1<br>Revision 4, September 2012. |
| [CC-2] | Common Criteria for Information Technology Security Evaluation<br>Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1<br>Revision 4, September 2012. |
| [CC-3] | Common Criteria for Information Technology Security Evaluation<br>Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1<br>Revision 4, September 2012. |
| [CCDB] | Common Criteria Supporting Document, Mandatory Technical Document – Composite product evaluation for Smart Cards and similar devices<br>Ref: CCDB-2012-04-001, Version 1.2, April 2012. |
| [PP-USIM] | (U)SIM Java Card Platform Protection Profile – Basic and SCWS Configurations<br>Ref: PU-2009-RT-79, Version 2.0.2, June 17th 2010 |
| [PP-JCS] | Java Card System Protection Profile – Open Configuration<br>Ref: ANSSI-CC-PP-2010/03, Version 2.6, April 19th 2010 |
| [PP/0035] | Security IC platform protection profile, version 1.0, 15th June 2007.<br>Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035. |
| [ST_ST33G1M2] | ST33G1M2 Platform Maskset K8H0A version F, with firmware revision 9, optional cryptographic library NESLIB 4.1, and optional technology MIFARE DESFire EV1 3.7 and 3.8 - Security Target - Public Version<br>Ref: SMD_ST33G_ST_13_002, Rev 02.03, June 2014, STMicroelectronics. |

## 1.2. INTERNAL REFERENCES [IR]

| [AGD] | TOE guidance documentation |
|---|---|
| [PRE] | Upteq NFC 3.X platform – Preparation Guidance<br>Ref: D1351549, version 1.1, Gemalto |
| [OPE_wCA] | Guidance for administration of Upteq NFC 3.X platform with Controlling Authority and Optional Verification Authority<br>Ref: D1341170_w_CA, version 1.1, Gemalto |
| [OPE_woCA] | Guidance for administration of Upteq NFC 3.X platform without Controlling Authority and Optional Verification Authority<br>Ref: D1341170_wo_CA, version 1.1, Gemalto |
| [OPE_VA] | Guidance for Verification Authority for Upteq NFC 3.X platform<br>Ref: D1341169_VA, version 1.0, Gemalto |
| [GUI_BasicApp] | GlobalPlatform Card – Composition Model – Security Guidelines for Basic Applications<br>Ref: GPC_GUI_050, version 2.0, November 2014, GlobalPlatform |
| [GUI_SecureApp] | Guidance for secure application development on Upteq NFC platforms<br>Ref: D1188231, release A13.1, Gemalto |
| [App_Mngt] | Application management for certified Secure Elements – External Procedure<br>Ref: D1258682, release C01, Gemalto |
| [PatchLoad_Mngt] | Patch Loading Management for certified Secure Elements – External procedure<br>Ref: D1344508, release A00, Gemalto |

# 2. Acronyms

| | |
|---|---|
| AP | Application Provider |
| API | Application Programming Interface |
| APSD | Application Provider Security Domain |
| CA | Controlling Authority |
| CASD | Controlling Authority Security Domain |
| CC | Common Criteria |
| CLF | Contactless Front-End |
| DAP | Data Authentication Pattern |
| EAL | Evaluation Assurance Level |
| GASD | GemActivate Security Domain |
| GP | GlobalPlatform |
| ISD | Issuer Security Domain |
| M4M | MIFARE For Mobile |
| MNO | Mobile Network Operator |
| NAA | Network Authentication Application |
| NFC | Near Field Communication |
| OTA | Over-The-Air |
| PP | Protection Profile |
| RAM | Random Access Memory |
| SD | Security Domain |
| ST | Security Target |
| SWP | Single Wire Protocol |
| TOE | Target Of Evaluation |
| TSM | Trusted Service Manager |
| USIM | Universal Subscriber Identity Module |
| VA | Verification Authority |
| VASD | Verification Authority Security Domain |
| VC | Virtual Card |

Copyright Gemalto SA – 2016

# 3. Security Target introduction

## 3.1. SECURITY TARGET IDENTIFICATION

| | |
|---|---|
| **Title:** | UpTeq NFC3.2.2_Generic v1.0 USIM Platform – Security Target |
| **Version:** | 1.0p |
| **Author:** | Gemalto |
| **Reference:** | D1350235 |
| **Publication date:** | 28/10/2015 |

## 3.2. TOE IDENTIFICATION

| | |
|---|---|
| **Product name:** | UpTeq NFC3.2.2_Generic v1.0 smartcard |
| **Product reference:** | T1032507 Release A |
| **TOE name:** | USIM platform part of the UpTeq NFC3.2.2_Generic v1.0 smartcard software |
| **TOE version:** | S1164861 Release A |
| **TOE documentation:** | Guidance [AGD] |
| **TOE hardware part:** | STMicroelectronics ST33G1M2 security controller |
| **Developer:** | Gemalto |

## 3.3. TOE OVERVIEW

The product UpTeq NFC3.2.2_Generic v1.0 is a UICC smartcard defined to be used in a mobile or smartphone. As such, it ensures the authentication of the subscriber to the MNO network, giving access to MNO services and applications.

It is also a multi-applicative security device intended to host payment, access control, transport or loyalty applications.

UpTeq NFC3.2.2_Generic v1.0 implements the [ISO7816] T=0 and Single Wire Protocol [TS 102.613] contact protocols. Inserted in a NFC-enabled mobile phone, it allows contactless communication with a terminal using the standard [ISO14443] communication protocol, on top of legacy ISO communication with the baseband. Thus, it offers convergence between the mobile communication environment and the convenience and security of contactless transactions based on smartcard technology.

Thus, a mobile NFC payment transaction is achieved by swiping the mobile over a NFC reader at a point of sale, creating a secure connection between the reader and the UICC smartcard in which a banking application secures the transaction.

In the same manner, a public transport access is granted to a user by swiping the mobile over a NFC reader, creating a secure connection between the reader and the UICC smartcard in which a transport application manages the access control operation.

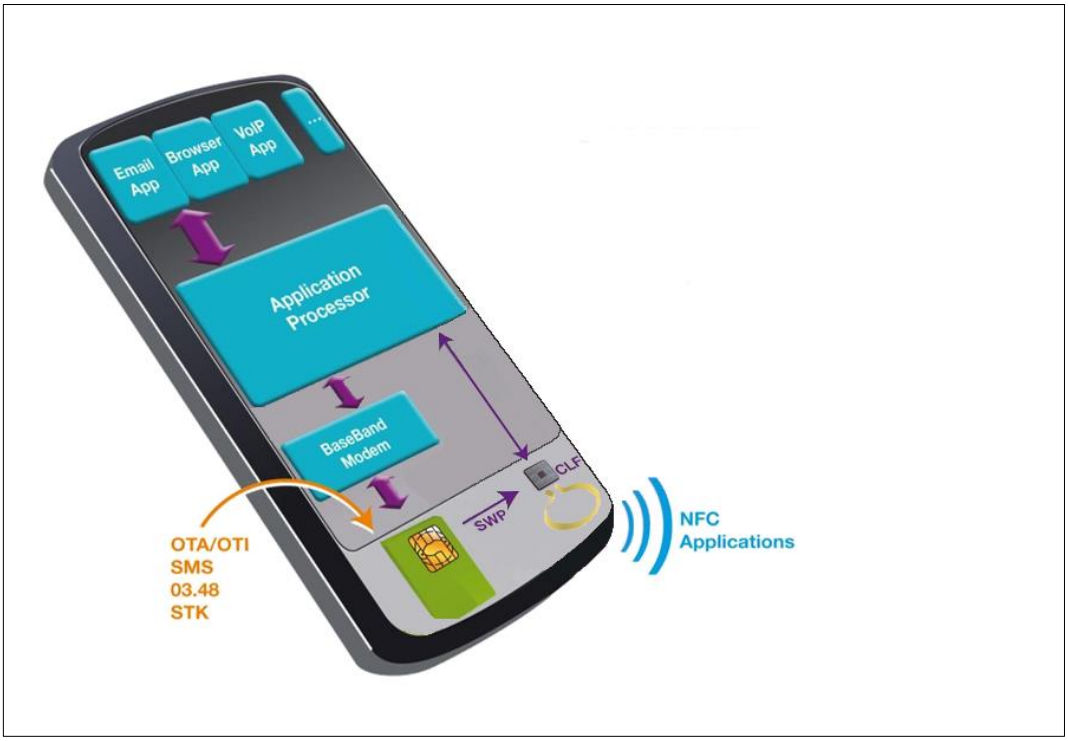


***Figure 1: TOE product environment***

UpTeq NFC3.2.2_Generic v1.0 is built upon an opened platform implementing the [Javacard] and Global Platform [GP] standards. Remote administration and post-issuance of applications is managed securely by the MNO and accredited TSM actors.

For the present evaluation, the Target of Evaluation (TOE) is the USIM platform part of the UpTeq NFC3.2.2_Generic v1.0 smartcard software. The TOE boundaries encompass:

- **The Javacard System (JCS)** implemented according to the [Javacard] standard, which manages and executes applications called applets. It also provides Javacard APIs for applet development
- **The GlobalPlatform (GP) packages** implemented according to the [GP] standard, which provide a common and widely used interface to communicate with a smartcard and manage applications in a secure way
- **The Telecom environment** implemented according to [ETSI] and [3GPP], including Network Authentication Applications (NAA, not evaluated) and Telecom communication protocols. It also provides [TS 31.130] APIs to specifically interact with USIM applications
- **The GemActivate application**, which is the Gemalto proprietary solution to activate services and/or load software patches post-issuance, under MNO and Gemalto administration
- **The ST33G1M2 Integrated Circuit**
- **The guidance documentation [AGD]**

Copyright Gemalto SA – 2016

# 4. TOE Description

## 4.1. ARCHITECTURE OF THE UICC SMARTCARD

The high-level architecture of the UpTeq NFC3.2.2_Generic v1.0 UICC smartcard can be represented by Figure 2. In this figure, the elements in blue are configurable.

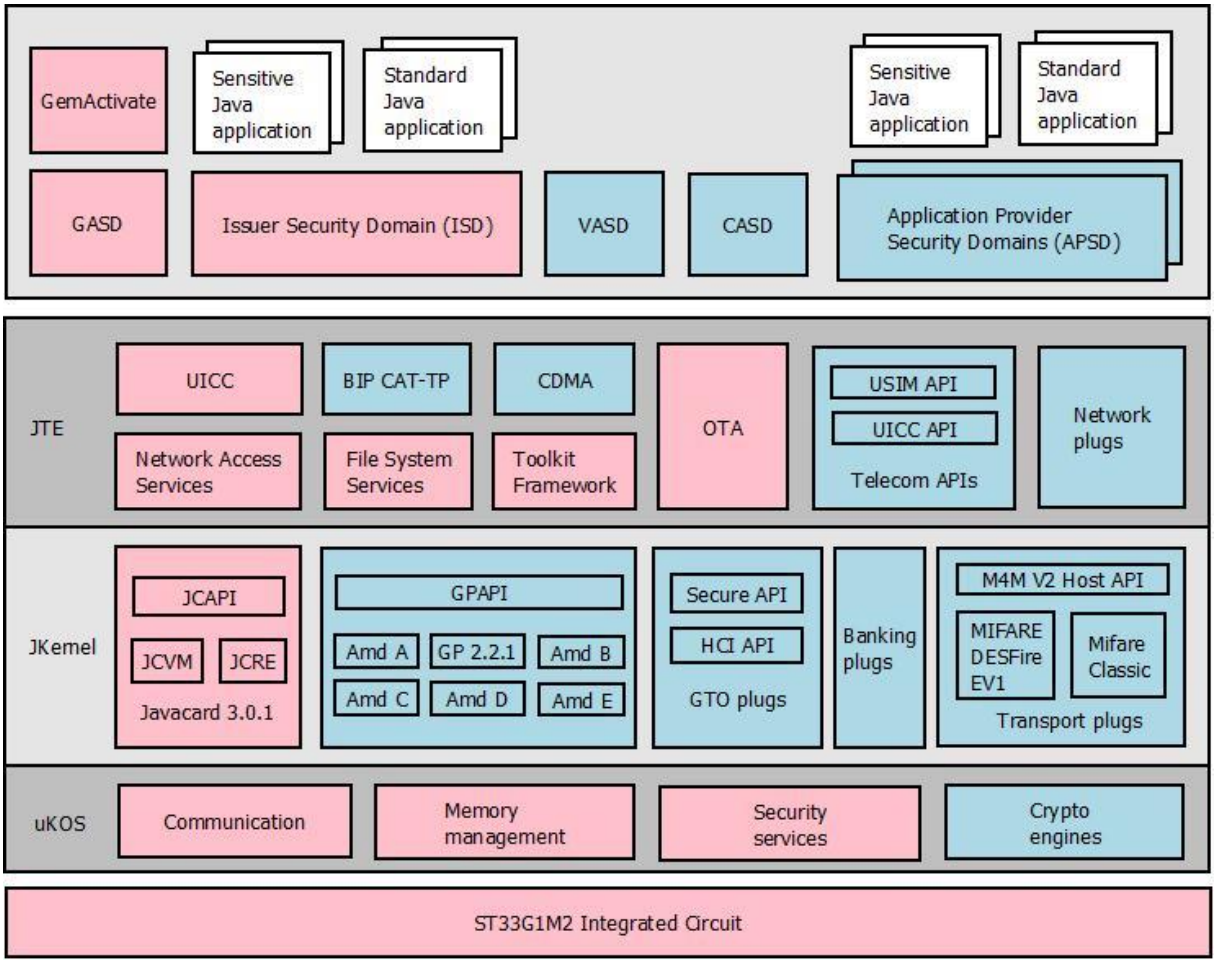


***Figure 2: UpTeq NFC3.2.2_Generic v1.0 smartcard architecture***

The architecture can be decomposed in three layers:

- The hardware layer composed of the ST33G1M2 integrated circuit
- The UpTeq NFC3.2.2_Generic v1.0 platform, which is the operating system of the product
- The application layer, encompassing standard and sensitive applications, as well as the security domains (ISD, GASD, VASD, CASD and APSDs).

## 4.2. TOE BOUNDARIES

### *4.2.1. TOE physical boundaries*

The UpTeq NFC3.2.2_Generic v1.0 UICC smartcard is a manufactured product composed of:

- The ST33G1M2 security controller with Gemalto NFC_3.2.2 v1.0 embedded software
- The smartcard module, which provides galvanic plates for electrical interface, and ensures the mechanical protection of the security controller
- The smartcard plastic body (usually in 2FF, 3FF or 4FF format, for plugging into the mobile phone)

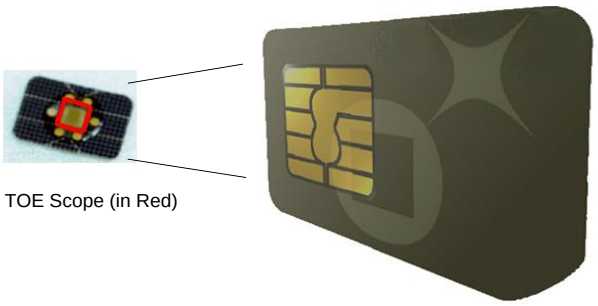


***Figure 3: TOE physical boundaries***

For the present evaluation, the TOE physical boundaries encompass the ST33G1M2 security controller with the Gemalto UpTeq NFC3.2.2_Generic v1.0 embedded software. The other smartcard items (smartcard module and plastic body, graphical printing...) are outside the scope of the evaluation, as illustrated in Figure 3.

### *4.2.2. TOE logical boundaries*

The present Security Target claims conformance to the [PP-USIM] Basic Configuration protection profile; the TOE logical boundaries are delimited (dash line in red) in Figure 4.

In this figure, the TSF components have been put in yellow color. The other components (in white color) do not participate to the TOE security.

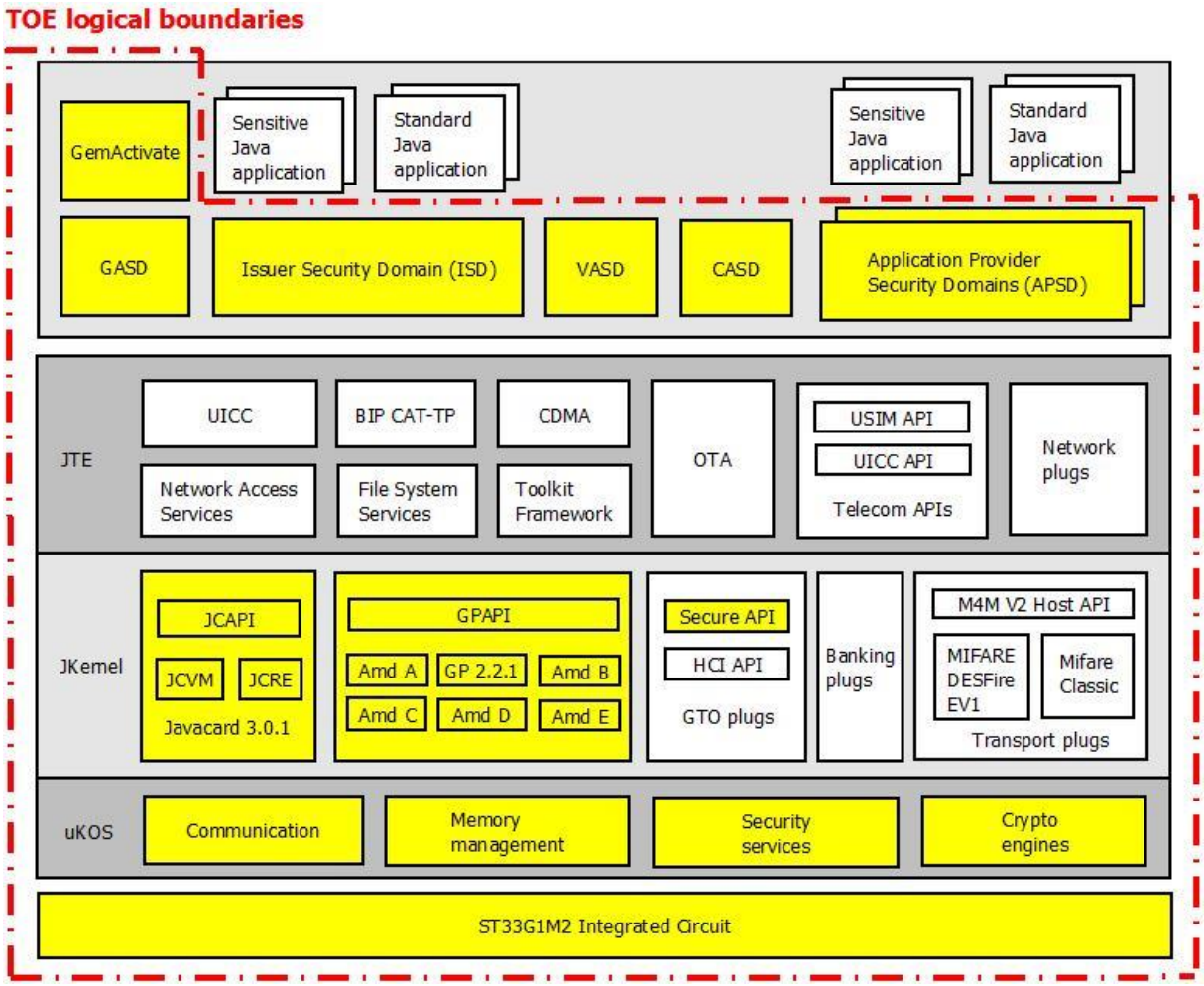


***Figure 4: TOE logical boundaries***

## 4.3. UPTEQ NFC3.2.2_GENERIC V1.0 PLATFORM DESCRIPTION

The UpTeq NFC3.2.2_Generic v1.0 platform implements two major industry standards:

- Oracle's Java Card 3.0.1 [Javacard], which consists of the Java Card 3.0.1 Virtual Machine, Java Card 3.0.1 Runtime Environment and the Java Card 3.0.1 Application Programming Interface.
- Global Platform 2.2.1 [GP], UICC Configuration.

It is an opened platform, meaning that additional applications - which may not be known at the time of the present evaluation - can be remotely loaded and installed "post issuance", i.e. after the UICC card has been delivered to the end-user. Applications can also be installed "pre-issuance" during the pre-personalization or personalization phases. Whatever the scenario (pre-issuance or post-issuance), applications' loading and installation are secured by the GlobalPlatform security mechanisms and verification processes.

The platform implements (at least) the following services:

- Management and control of the communication between the card and external entities

- Card basic security services as follows:
  - Checking environmental operating conditions using information provided by the IC,
  - Checking life cycle consistency,
  - Providing secure cryptography primitives and algorithms
  - Ensuring the security of the PIN and cryptographic key objects
  - Generating random numbers,
  - Handling secure data object and backup mechanisms,
  - Managing memory content,
- Enforcement of the Javacard firewall mechanism
- Standard Application Programming Interfaces (APIs) such as the Javacard API (JCAPI) and the Global Platform API (GPAPI)
- Proprietary Gemalto API: Secure API which provides security services to applications
- Initialization of the Issuer Security Domain (ISD) and management of the card life cycle
- Creation and management of Supplementary Security Domains (SSD)
- SCP02, SCP03, SCP80, SCP81 support
- RSA, ECC support
- Secure loading, installation and deletion of applications within each SD
- Extradition
- DAP and mandated DAP support
- Delegated Management privilege
- Trusted Path privilege
- Secure loading of software patches (GemActivate)

Note: the GemActivate application is associated by default to ISD. GASD is optional and created only on MNO decision. In such case, GemActivate application can be extradited on GASD to use dedicated SCP80 secure channel. For ETSI secure scripting according to GP UICC configuration, by default ISD SCP80 is used, otherwise GASD or ascendant SD of GASD (e.g. ISD) SCP80 is used. In both cases, GemActivate application performs applicative checks prior any required operation.

Note: The Authorized Management privilege is only supported for ISD as excluded in [PP-USIM].

The UpTeq NFC3.2.2_Generic v1.0 platform implements MIFARE DESFire EV1 and MIFARE4MOBILE V2 according to the NXP specifications [MIFARE]. MIFARE DESFire EV1 targets secure contactless transport applications and related loyalty programs to which it provides a set of dedicated services; it is included within the TOE but doesn't provide any security function for the present evaluation.

As part of a UICC product, the UpTeq NFC3.2.2_Generic v1.0 platform also implements a Javacard Telecom Environment (JTE) compliant with the [ETSI] and [3GPP] specifications. Among other features, the JTE supports one or several Network Authentication Applications (NAA), file system management, SIM toolkit, OTA and BIP functionalities, as well as USIM/UICC APIs. As shown in Figure 4, the JTE is included within the TOE but doesn't provide any security function for the present evaluation.

## 4.4. APPLICATION LAYER DESCRIPTION

Applications can be split in two categories:

- Secure applications: these are sensitive applications, such as e.g. banking applets, whose security is assessed and certified through international schemes (Common Criteria, EMVco etc.)
- Standard applications, also called "basic" applications: these are the other applications. Although they do not face a formal security evaluation, assurance has to be provided that they do not threaten the sensitive applications and their assets. This assurance is provided through a verification process. Security mechanisms are in

place at platform level to ensure that applications which are loaded post issuance have been verified.

## 4.5. TOE LIFE-CYCLE

The product and TOE life cycle is composed of 7 phases which are described in table 1. The table also mentions the actor(s) involved in each phase, as well as the associated location(s).

The IC does not contain any part of the UpTeq NFC3.2.2_Generic v1.0 software prior to phase 6. The loading of the UpTeq NFC3.2.2_Generic v1.0 software occurs during phase 6, after which the IC loading service is locked and no more available.

As described, at the end of phase 6 Gemalto delivers personalized UpTeq NFC3.2.2_Generic v1.0 cards to the Mobile Network Operator (MNO). At this stage, the TOE is entirely built and protects itself through the security mechanisms implemented in the operating system and the underlying IC. Consequently, the TOE delivery point - which determines the boundary between the ALC and AGD Common Criteria assurance classes - is put at the end of phase 6, as illustrated in figure 5.

Notes related to applications development

The basic and secure applets development is part of the product life cycle, but is outside the scope of the present evaluation (since applications are out of the TOE).

The Gemalto applications will be verified using the evaluated Gemalto verification process prior to be loaded in Pre-Issuance by Gemalto.

In the same way, but to protect the supplier intellectual property, the applications provided by external Application Providers must be verified and signed by the Verification Authority (VA) prior to be loaded in Pre-Issuance by Gemalto. Gemalto will check application signature prior to load these applications in Pre-Issuance.

Note related to patch development

No patch is present within the TOE for the present evaluation. Indeed, should a patch be needed in the future, it would require at least a maintenance of the CC certificate, as required by the CC scheme rules. However the patch mechanism is part of the TOE and as such its security is assessed within the present evaluation.

| Phase | Designation | Description / comments | | Actor | Location |
|---|---|---|---|---|---|
| 1 | UpTeq NFC3.2.2_Generic v1.0 software development | UpTeq NFC3.2.2_Generic v1.0 platform development | Platform development & tests | **Gemalto** Telecom R&D team<br>- secure environment - | **Gemalto La Ciotat** site |
| | | | | **Gemalto** SL Crypto team<br>- secure environment - | **Gemalto Singapore** site |
| | | Patch development | Patch development and tests | **Gemalto**<br>Telecom R&D and SL Crypto teams<br>- secure environment - | **Gemalto La Ciotat** site<br>**Gemalto Singapore** site |
| | | Basic and secure applets development | Applet development & tests | **Gemalto or any other accredited Application Provider (AP)**<br>- secure environment - | Application Providers' development sites |
| | | Industrialization | Management of the delivery process between Gemalto and STMicroelectronics | **Gemalto** Component group<br>- secure environment - | **Gemalto Gémenos** site |
| | | | Production scripts development for phases 4 and 5 (assembly, preperso). Delivery to the production sites. | **Gemalto** PSE team<br>- secure environment - | **Gemalto Gémenos** site<br>**Gemalto Singapore** site |
| | | | Perso scripts development for phase 6. Delivery to the personalization sites. | **Gemalto** CPC team<br>- secure environment - | **Gemalto Tczew** site |
| 2 | IC development | Development of the ST33G1M2 security controller and associated tools. | | **STMicroelectronics**<br>- Secure environment - | STMicroelectronics development site(s) as stated in the ST33G1M2 CC certificate |
| 3 | IC manufacturing | Manufacturing of virgin ST33G1M2 integrated circuits embedding the STMicroelectronics flash loader, and protected by a dedicated Gemalto key. | | **STMicroelectronics**<br>- Secure environment - | STMicroelectronics development site(s) as stated in the ST33G1M2 CC certificate |
| 4 | Module manufacturing | IC packaging & testing | | **Gemalto**<br>- Secure environment - | **Gemalto Pont-Audemer** site<br>**Gemalto Tczew** site<br>**Gemalto Singapore** site<br>**Gemalto Curitiba** site |
| 5 | Card manufacturing and pre-personalization | Module embedding in plastic card body<br>Preperso and Testing | | **Gemalto**<br>- Secure environment - | **Gemalto Pont-Audemer** site<br>**Gemalto Tczew** site<br>**Gemalto Singapore** site<br>**Gemalto Curitiba** site |

Copyright Gemalto SA – 2016

| Phase | Designation | Description / comments | Actor | Location |
|---|---|---|---|---|
| 6 | UpTeq NFC3.2.2_Generic v1.0 card personalization | Loading of the NFC3.2.2_Generic v1.0 software (platform and pre-issuance applications). Creation of files and loading of end-user data. | **Gemalto**<br>- Secure environment - | **Gemalto Pont-Audemer** site<br>**Gemalto Tczew** site<br>**Gemalto Singapore** site<br>**Gemalto Curitiba** site |
| 7 | End-usage | End-usage for the Mobile Network Operator (MNO) and accredited business partners (Application Providers).<br><br>The MNO, who is the issuer of the UpTeq NFC3.2.2_Generic v1.0 smartcard, is responsible for the card administration during the end-usage phase and the end of life process. The MNO also grants administration privileges to Application Providers on their respective Security Domains (APSD). | **Mobile Network Operator and accredited business partners (Application Providers)** | Field |
| | | End-usage for mobile phone holder<br><br>The end-user accesses the MNO network and related services, and performs secure NFC transactions with his mobile phone, thanks to the UpTeq NFC3.2.2_Generic v1.0 smartcard hosting the sensitive applications and related assets. | **Mobile phone holder** | Field |

***Table 1: Product and TOE life-cycle phases***

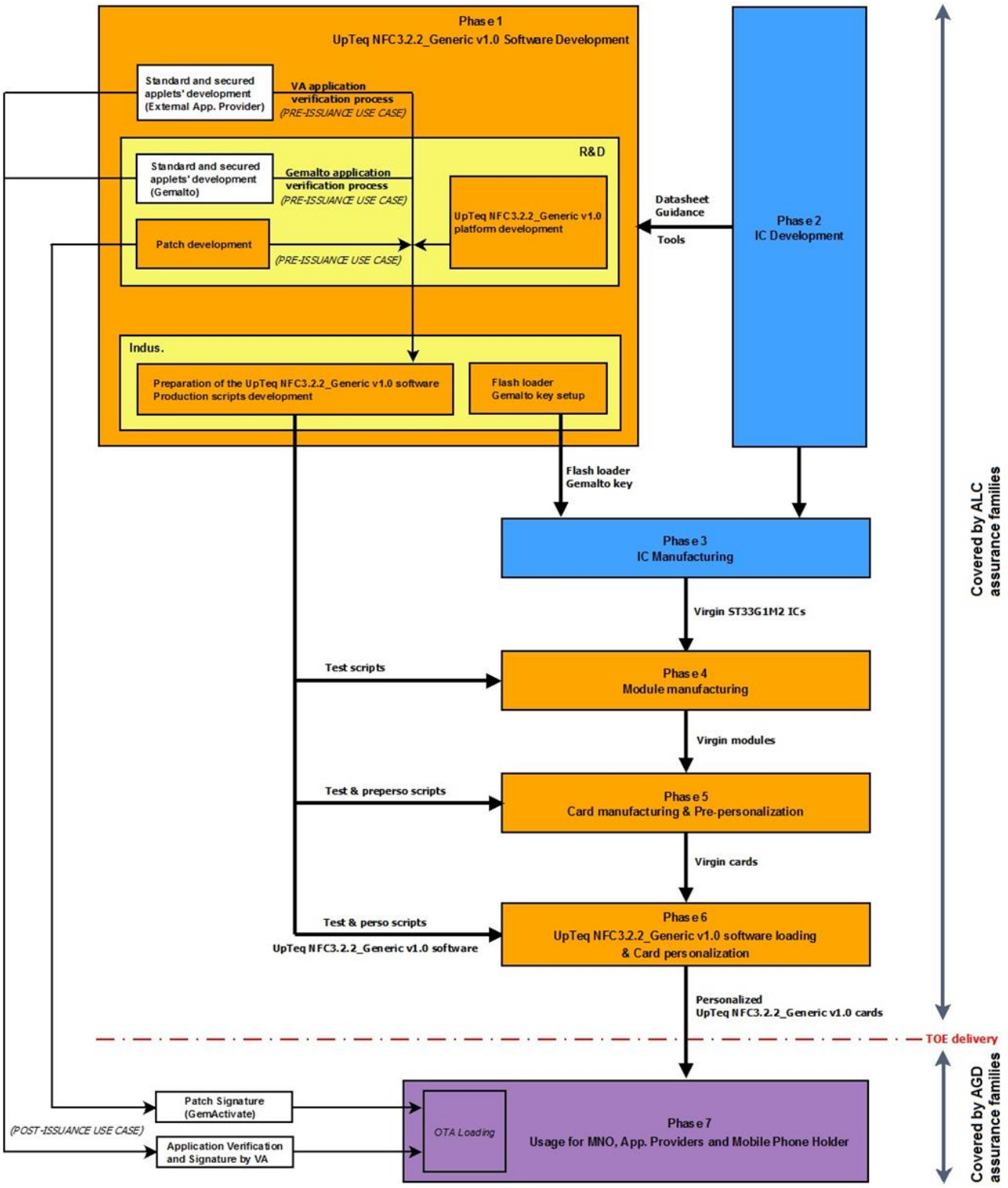


***Figure 5: Product and TOE life-cycle***

## 4.6. TOE ACTORS

The following actors are represented within the TOE:

- ***The Mobile Network Operator (MNO)***, who is the issuer of the UpTeq NFC3.2.2_Generic v1.0 smartcard and owner of the TOE. The TOE authorizes the MNO, once authenticated, to manage the loading, instantiation or deletion of applications.
- ***The Application Providers (AP)*** are entities or institutions responsible for their applications and associated services. It may be for example a financial institution (a bank) or a transport operator.
- ***The Controlling Authority (CA)***, optional entity independent from the MNO represented on the TOE and responsible for securing the keys creation and personalization of the Application Provider Security Domains (APSD).
- ***The Verification Authority (VA)***, trusted third party represented on the TOE, acts on behalf of the MNO and is responsible for the verification of applications signatures (mandated DAP) during the loading process. These applications shall be validated for the standard applications or certified for the secure ones.
- ***The GemActivate Administrator*** (usually Gemalto), represented on the TOE by the GemActivate application and associated keys, is responsible for the remote installation of platform patches (if needed) and the activation of optional platform services on the field (post-issuance).

Copyright Gemalto SA – 2016

# 5. Conformance claims

**Common criteria Version:** This ST conforms to CC Version 3.1 [CC-1] [CC-2] [CC-3]

**Conformance to CC part 2 and 3:**

- This ST is CC part 2 extended with the FCS_RND.1 family. All the other SFRs have been drawn from the catalogue of requirements in CC part 2 [CC-2].
- This ST is CC part 3 conformant. It means that all SARs in that ST are based only upon assurance components in CC part 3 [CC-3].

**Assurance package conformance:** EAL4 augmented (EAL4+)
This ST conforms to the assurance package EAL4 augmented by ALC_DVS.2 and AVA_VAN.5.

**Evaluation type**
This is a composite evaluation, which relies on the ST33G1M2 chip certificate and evaluation results.

**ST33G1M2 chip certificate:**

- Certification done under the ANSSI scheme
- Certification report ANSSI-CC-2014/46
- Security Target [ST_ST33G1M2] strictly conformant to IC Protection Profile [PP/0035]
- Common criteria version: 3.1
- Assurance level: EAL5+ (ALC_DVS.2 and AVA_VAN.5 augmentations)

Consequently, the composite product evaluation (i.e. the present evaluation) includes the additional composition tasks defined in the CC supporting document “Composite product evaluation for smart cards and similar devices” [CCDB].

**Protection Profile (PP) conformance claim:**
This Security Target claims conformance to the [PP-USIM] protection profile, Basic Configuration. The conformance type is demonstrable.

# 6. Security problem definition

## 6.1. ASSETS

### *6.1.1. USIM Protection Profile*

The following assets are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| D.ISD_KEYS | Issuer Security Domain cryptographic keys needed to perform card management operations on the card.<br>To be protected from unauthorized disclosure and modification. |
| D.APSD_KEYS | Application Provider Security Domains cryptographic keys needed to establish secure channels with the AP. These keys can be used to load and install applications on the card if the Security Domain has the appropriate privileges.<br>To be protected from unauthorized disclosure and modification. |
| D.CASD_KEYS | Controlling Authority Security Domains cryptographic keys needed to establish secure channels with the CA and to decrypt confidential content for APSDs.<br>To be protected from unauthorized disclosure and modification. |
| D.VASD_KEYS | Verification Authority Security Domain cryptographic keys needed to verify applications Mandated DAP signature.<br>To be protected from unauthorized disclosure and modification. |
| D.(U)SIM_DATA | Private data of the (U)SIM application, like the contents of its private fields.<br>To be protected from unauthorized disclosure and modification. |
| D.(U)SIM_CODE | The code of the (U)SIM application on the card.<br>To be protected from unauthorized modification. |
| D.GP_CODE | The code of the GlobalPlatform framework on the card.<br>To be protected from unauthorized modification. |
| D.CARD_MNGT_DATA | The data of the card management environment, like for instance, the identifiers, the privileges, life cycle states, the memory resource quotas of applets and security domains.<br>To be protected from unauthorized modification. |

### *6.1.2. JCS Protection Profile*

The following assets are listed in [PP-JCS]. According to [PP-USIM] they shall also be considered for the present evaluation.

| | |
|---|---|
| D.APP_CODE | The code of the applets and libraries loaded on the card.<br>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.<br>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.<br>To be protected from unauthorized modification. |
| D.APP_KEYS | Cryptographic keys owned by the applets.<br>To be protected from unauthorized disclosure and modification. |
| D.PIN | Any end-user's PIN.<br>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.CRYPTO | Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key.<br>To be protected from unauthorized disclosure and modification. |
| D.JCS_CODE | The code of the Java Card System.<br>To be protected from unauthorized disclosure and modification |
| D.JCS_DATA | The internal runtime data areas necessary for the execution of the Java Card VM, such as, for instance, the frame stack, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures.<br>To be protected from unauthorized disclosure or modification. |
| D.SEC_DATA | The runtime security data of the Java Card RE, 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.<br>To be protected from unauthorized disclosure and modification. |

### *6.1.3. Supplementary assets*

The following assets are related to optional service activation and patch management in post-issuance phase (phase 7). As mentioned in section 4.5, there is no patch associated to the present TOE, however the patch and optional service activation mechanisms are within the evaluation scope.

| | |
|---|---|
| D.GASD_KEYS | Gemalto Security Domain cryptographic keys needed to authorize activation and patch loading requests.<br>To be protected from unauthorized disclosure and modification. |
| D.OPTIONAL_PF_SERVICE | Platform services can be configured by addition of optional services as e.g.<br>- new cryptographic algorithm service available through API<br>- new network authentication algorithm available through API<br>To be protected from unauthorized modification. |
| D.PF_Additional_CODE | It is the code (javacard package) to be added to the platform by the platform developer after TOE issuance. Code has to be loaded and installed on the TOE through an atomic activation (to create an updated TOE).<br>To be protected from unauthorized modification. |
| D.PF_Additional_CODE_Identifier | It is an identifier associated to the additional code. The identifier is loaded in the same atomic operation than additional code loading.<br>To be protected from unauthorized modification. |
| D.PF_Additional_CODE_Certificate | Certificate generated by the platform developer, which allows the TOE to check the authenticity and integrity of the Additional Code.<br>To be protected from unauthorized modification. |

## 6.2. USERS / SUBJECTS

Subjects are active components of the TOE that (essentially) act on the behalf of users. Users of the TOE include people or institutions (like the AP, the MNO and the VA), hardware (like the CAD where the card is inserted) and software components (like the application packages installed on the card).

In this Security Target, relevant subjects are those listed in [PP-JCS] plus the following ones:

| | |
|---|---|
| S.SD | A GlobalPlatform Security Domain representing on the card an off-card entity. This entity can be the Issuer, an Application Provider, the Controlling Authority or the Validation Authority. |

| | |
|---|---|
| S.GEMACTIVATE | GemActivate Security Domain representing on the card a Gemalto administrator. This entity can authorize the activation of optional services and the loading of additional code (i.e. patch) post issuance. |

## 6.3. THREATS

### *6.3.1. USIM Protection Profile*

The following assets are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| 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. This threat includes IC failure analysis, electrical probing, unexpected tearing, and DPA. That also includes the modification of the TOE runtime execution through alteration of the intended execution order of (set of) instructions through physical tampering techniques.<br>This threatens all the identified assets. |
| T.INTEG-USER-DATA | The attacker through a malicious applet loaded on the card modifies application data, application keys or authentication data.<br>Directly threatened asset(s): D.(U)SIM_DATA, D.ISD_KEYS, D.VASD_KEYS, D.APSD_KEYS and D.CASD_KEYS. |
| T.COM_EXPLOIT | An attacker remotely exploits the communication channel (USB, ISO-7816, NFC, BIP or SMS) established between the mobile phone and the (U)SIM card in order to modify or disclose confidential data.<br>All assets are threatened. |
| T.UNAUTHORIZED_CARD_MNGT | The attacker performs unauthorized card management operations (for instance impersonates one of the actor represented on the card) in order to take benefit of the privileges or services granted to this actor on the card such as fraudulent:<br>▪ load of a package file<br>▪ installation of a package file<br>▪ extradition of a package file or an applet<br>▪ personalization of an applet or a Security Domain<br>▪ deletion of a package file or an applet<br>▪ privileges update of an applet or a Security Domain<br>Directly threatened asset(s): D.ISD_KEYS, D.CASD_KEYS, D.APSD_KEYS, D.APP_C_DATA (from [PP-JCS]), D.APP_I_DATA (from [PP-JCS]), D.APP_CODE (from [PP-JCS]) and D.CARD_MNGT_DATA. |
| T.LIFE_CYCLE | An attacker accesses to an application outside of its expected availability range thus violating irreversible life cycle phases of the application (for instance, an attacker re-personalizes the application).<br>Directly threatened asset(s): D.APP_I_DATA (from [PP-JCS]), D.APP_C_DATA (from [PP-JCS]), and D.CARD_MNGT_DATA. |
| T.UNAUTHORIZED_ACCESS | By using the shareable object mechanism on which relies the communication between two applets, the attacker uses an applet on card to get access or to modify data from another applet that he should not have access to.<br>All assets are threatened. |

### 6.3.2. JCS Protection Profile

According to [PP-USIM], the threats listed in [PP-JCS] shall also be considered for the present evaluation. The following table gathers elements extracted from [PP-JCS] which will be referred to in some of the threats mentioned in this section.

| | |
|---|---|
| *#.CONFID-APPLI-DATA* | *Application data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain read access to other application's data.* |
| *#.CONFID-JCS-CODE* | *Java Card System code must be protected against unauthorized disclosure. Knowledge of the Java Card System code may allow bypassing the TSF. This concerns logical attacks at runtime in order to gain a read access to executable code, typically by executing an application that tries to read the memory area where a piece of Java Card System code is stored.* |
| *#.CONFID-JCS-DATA* | *Java Card System data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain a read access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card platform API classes as well.* |
| *#.INTEG-APPLI-CODE* | *Application code must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to the memory zone where executable code is stored. In post-issuance application loading, this threat also concerns the modification of application code in transit to the card.* |
| *#.INTEG-APPLI-DATA* | *Application data must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain unauthorized write access to application data. In post-issuance application loading, this threat also concerns the modification of application data contained in a package in transit to the card. For instance, a package contains the values to be used for initializing the static fields of the package.* |
| *#.INTEG-JCS-CODE* | *Java Card System code must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to executable code.* |
| *#.INTEG-JCS-DATA* | *Java Card System data must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card API classes as well.* |
| *#.EXE-APPLI-CODE* | *Application (byte)code must be protected against unauthorized execution. This concerns (1) invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([JAVASPEC], §6.6); (2) jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code; (3) unauthorized execution of a remote method from the CAD (if the TOE provides JCRMI functionality).* |
| *#.EXE-JCS-CODE* | *Java Card System bytecode must be protected against unauthorized execution. Java Card System bytecode includes any code of the Java Card RE or API. This concerns (1) invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([JAVASPEC], §6.6); (2) jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code. Note that execute access to native code of the Java Card System and applications is the concern of #.NATIVE.* |
| *#.FIREWALL* | *The Firewall shall ensure controlled sharing of class instances7, and isolation of their data and code between packages (that is, controlled execution contexts) as well as between packages and the JCRE context. An applet shall not read, write, compare a piece of data belonging to an applet that is not in the same context, or execute one of the methods of an applet in another context without its authorization.* |
| *#.NATIVE* | *Because the execution of native code is outside of the JCS TSF scope, it must be secured so as to not provide ways to bypass the TSFs of the JCS. Loading of native code, which is as well outside those TSFs, is submitted to the same requirements. Should native software be privileged in this respect, exceptions to the policies must include a rationale for the new security framework they introduce.* |
| *#.VERIFICATION* | *Bytecode must be verified prior to being executed. Bytecode verification includes (1) how well-formed CAP file is and the verification of the typing constraints on the bytecode, (2) binary compatibility with installed CAP files and the assurance that the export files used to check the CAP file correspond to those that will be present on the card when loading occurs.* |
| *#.INSTALL* | *(1) The TOE must be able to return to a safe and consistent state when the installation of a package or an applet fails or be cancelled (whatever the reasons). (2) Installing an applet must have no effect on the code and data of already installed applets. The installation procedure should not be used to bypass the TSFs. In short, it is an atomic* |

| | |
|---|---|
| | *operation, free of harmful effects on the state of the other applets. (3) The procedure of loading and installing a package shall ensure its integrity and authenticity.* |
| *#.SID* | *(1) Users and subjects of the TOE must be identified. (2) The identity of sensitive users and subjects associated with administrative and privileged roles must be particularly protected; this concerns the Java Card RE, the applets registered on the card, and especially the default applet and the currently selected applet (and all other active applets in Java Card System 2.2.x). A change of identity, especially standing for an administrative role (like an applet impersonating the Java Card RE), is a severe violation of the Security Functional Requirements (SFR). Selection controls the access to any data exchange between the TOE and the CAD and therefore, must be protected as well. The loading of a package or any exchange of data through the APDU buffer (which can be accessed by any applet) can lead to disclosure of keys, application code or data, and so on.* |
| *#OBJ-DELETION* | *(1) Deallocation of objects should not introduce security holes in the form of references pointing to memory zones that are no longer in use, or have been reused for other purposes. Deletion of collection of objects should not be maliciously used to circumvent the TSFs. (2) Erasure, if deemed successful, shall ensure that the deleted class instance is no longer accessible.* |
| *#DELETION* | *(1) Deletion of installed applets (or packages) should not introduce security holes in the form of broken references to garbage collected code or data, nor should they alter integrity or confidentiality of remaining applets. The deletion procedure should not be maliciously used to bypass the TSFs. (2) Erasure, if deemed successful, shall ensure that any data owned by the deleted applet is no longer accessible (shared objects shall either prevent deletion or be made inaccessible). A deleted applet cannot be selected or receive APDU commands. Package deletion shall make the code of the package no longer available for execution. (3) Power failure or other failures during the process shall be taken into account in the implementation so as to preserve the SFRs. This does not mandate, however, the process to be atomic. For instance, an interrupted deletion may result in the loss of user data, as long as it does not violate the SFRs.* |
| *#.RESOURCES* | *The TOE controls the availability of resources for the applications and enforces quotas and limitations in order to prevent unauthorized denial of service or malfunction of the TSFs. This concerns both execution (dynamic memory allocation) and installation (static memory allocation) of applications and packages.* |

The following threats are derived from the here-above security aspects:

| | |
|---|---|
| T.CONFID-APPLI-DATA | The attacker executes an application to disclose data belonging to another application. See #.CONFID-APPLI-DATA for details.<br>Directly threatened asset(s): D.APP_C_DATA and D.APP_KEYs. |
| T.CONFID-JCS-DATA | The attacker executes an application to disclose data belonging to the Java Card System. See #.CONFID-JCS-DATA for details.<br>Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO. |
| T.CONFID-JCS-CODE | The attacker executes an application to disclose the Java Card System code. See #.CONFID-JCS-CODE for details.<br>Directly threatened asset(s): D.JCS_CODE. |
| T.INTEG-APPLI-CODE | The attacker executes an application to alter (part of) its own code or another application's code. See #.INTEG-APPLI-CODE for details.<br>Directly threatened asset(s): D.APP_CODE. |
| T.INTEG-APPLI-CODE.LOAD | The attacker modifies (part of) its own or another application code when an application package is transmitted to the card for installation. See #.INTEG-APPLI-CODE for details.<br>Directly threatened asset(s): D.APP_CODE. |
| T.INTEG-APPLI-DATA | The attacker executes an application to alter (part of) another application's data. See #.INTEG-APPLI-DATA for details.<br>Directly threatened asset(s): D.APP_I_DATA and D.APP_KEYs. |
| T.INTEG-APPLI-DATA.LOAD | The attacker modifies (part of) the initialization data contained in an application package when the package is transmitted to the card for installation. See #.INTEG-APPLI-DATA for details. |

| | |
|---|---|
| | Directly threatened asset(s): D.APP_I_DATA and D_APP_KEY. |
| T.INTEG-JCS-DATA | The attacker executes an application to alter (part of) Java Card System or API data. See #.INTEG-JCS-DATA for details.<br>Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO. |
| T.INTEG-JCS-CODE | The attacker executes an application to alter (part of) the Java Card System code. See #.INTEG-JCS-CODE for details.<br>Directly threatened asset(s): D.JCS_CODE. |

Other attacks are in general related to one of the above, and aimed at disclosing or modifying on-card information. Nevertheless, they vary greatly on the employed means and threatened assets, and are thus covered by quite different objectives. That is why a more detailed list is given hereafter.

| | |
|---|---|
| T.SID.1 | An applet impersonates another application, or even the Java Card RE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal. See #.SID for details.<br>Directly threatened asset(s): D.SEC_DATA (other assets may be jeopardized should this attack succeed, for instance, if the identity of the JCRE is usurped), D.PIN and D.APP_KEYs. |
| T.SID.2 | The attacker modifies the TOE's attribution of a privileged role (e.g. default applet and currently selected applet), which allows illegal impersonation of this role. See #.SID for further details.<br>Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged). |
| T.EXE-CODE.1 | An applet performs an unauthorized execution of a method. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details.<br>Directly threatened asset(s): D.APP_CODE. |
| T.EXE-CODE.2 | An applet performs an execution of a method fragment or arbitrary data. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details.<br>Directly threatened asset(s): D.APP_CODE. |
| T.NATIVE | An applet executes a native method to bypass a TOE Security Function such as the firewall. See #.NATIVE for details.<br>Directly threatened asset(s): D.JCS_DATA. |
| T.RESOURCES | An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM. See #.RESOURCES for details.<br>Directly threatened asset(s): D.JCS_DATA. |
| 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). See #.DELETION for details).<br>Directly threatened asset(s): D.SEC_DATA and D.APP_CODE. |
| T.INSTALL | The attacker fraudulently installs post-issuance of 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. See #.INSTALL for details.<br>Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on the virulence of the installed application). |
| 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. See #.OBJ-DELETION for further details.<br>Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYs. |

### *6.3.3. Supplementary threats*

The following threats are related to optional service activation and patch loading in post-issuance.

| | |
|---|---|
| T.UNAUTHORIZED_ACCESS_TO_ SERVICE | An attacker may gain direct access to an optional platform service without authorization by bypassing access control to service activation.<br><br>Directly threatened asset(s): D.GASD_KEYS, D.OPTIONAL_PF_SERVICE |
| T.UNAUTHORIZED_TOE_CODE_UPDATE | An attacker attempts to update the TOE code with a malicious update that may compromise the security features of the TOE.<br><br>Directly threatened asset(s): D.PF_Additional_CODE. |
| T.UNAUTHORIZED_TOE_CODE_UPDATE_BLOCK | An attacker attempts to continue to use the initial TOE without detection by blocking an authorized update of the TOE code to avoid improvement of the security features of the TOE.<br>Directly threatened asset(s): D.PF_Additional_CODE. |
| T.TOE_IMPERSONATION | An attacker attempts to update the TOE code without changing identifier or to update the identifier without changing the TOE code that may impersonate TOE and compromise the security features of the TOE.<br>Directly threatened asset(s): D.PF_Additional_CODE, D.PF_Additional_CODE_Identifier, D.PF_Additional_CODE_Certificate. |
| T.TOE_CLONING | An attacker attempts to download the updated TOE code and identifier in a fake TOE that may impersonate a real TOE and compromise the security features of the TOE.<br>Directly threatened asset(s): D.PF_Additional_CODE, D.PF_Additional_CODE_Identifier, D.PF_Additional_CODE_Certificate. |

## 6.4. ORGANISATIONAL SECURITY POLICIES

### *6.4.1. USIM Protection Profile*

The following OSP are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| OSP.SECURE-APPS-CERTIFICATION | Secure applications must be certified according to the Common Criteria at an EAL equal to the one of the current Protection Profile. The composition of these applications with the current PP must follow the rules defined in the document [Comp]. These applications are associated to a digital signature which will be checked by a VA during the loading into the TOE.<br>Application note: This composition process requires that platform administrator and user guides (AGD_ADM and AGD_USR) are available to the secure application developer. The Evaluation report for the composition (ETR-COMP), delivered by the ITSEF which manages applications composition, must be also provided. See [Secure APP] for more details on the evaluation/validation process. |
| OSP.BASIC-APPS-VALIDATION | Basic applications shall be associated to a digital signature which will be checked by a VA during the loading into the TOE. In addition to the rules stated by the Java Card specification, the validation process must enforce that basic applications:<br>▪ must follow the extra-rules stated in the user manual of the considered (U)SIM Java Card Platform,<br>▪ cannot be libraries,<br>▪ must not use RMI,<br>▪ must not use proprietary libraries which are not certified (except system libraries), |

| | |
|---|---|
| | ▪ access control to certified proprietary libraries is controlled by the secure application which has defined the library,<br>▪ must be associated to an identifier and this identifier has to be used in parameter of the function calls.<br><br>Application note: GSM file system and API's STK application descriptors are other ways to share object between applications. Identifier usage allows to easily track applications calls. This is useful if a new attack path is discovered to identify the pieces of code that could be vulnerable. See [Basic APP] for more details on the validation process. |
| OSP.SHARE-CONTROL | The Shareable interface functionality should be strictly controlled for all applications to prevent transitive data flows between applets (i.e., no re-sharing of a shareable object with a third applet) and thus prevent access to unauthorized data. |
| OSP.AID-MANAGEMENT | When loading an application that uses shareable object interface, to make its services available to other applications, the VA or the MNO shall verify that the AID of the application being loaded does not impersonate the AID known by another application on the card for the use of shareable services. |
| OSP.OTA-LOADING | Application code, validated or certified depending on the application, is loaded "Over The Air" (OTA) onto (U)SIM Platform using OTA servers of the mobile operator.<br><br>If needed, the Card issuer can pre-authorize content loading operation through delegated management privilege to individual on-card representative of APs. In that case the application code is loaded in the APSD.<br><br>Once loaded, the application is personalized using the appropriate SD keys. |
| OSP.OTA-SERVERS | A security policy shall be employed by the mobile operator to ensure the security of the applications stored on its servers.<br><br>Application note: The policy enforced by the mobile operator to ensure the security of the application can use mechanisms such as access control, isolation, regular check of integrity and encryption.<br>One possible realization of this Organizational Security Policy is the enforcement of security rules defined in OTA servers security guidance document with regular site inspections to check the applicability of the rules. |
| OSP.APSD-KEYS | The APSD keys personalization can rely either on the key escrow if the APSD has been created before the usage phase of the (U)SIM card or on the CA if the APSD has been created during the usage phase. In the first case, the security domain keys of the AP (APSD keys) are generated and stored in a secure way by the personalizer. Then, these keys are transmitted to the AP, via the key escrow, at the only mobile operator request. In the second case, the APSD keys are:<br>▪ Either generated and stored in a secure way by the APSD. Then these keys are securely transmitted to the AP using the CASD (Pull Model of [GP-CCCM]),<br>▪ Or created by the AP and securely transferred to the APSD using the CASD (Push Model of [GP-CCCM]).<br>Generated keys must be unpredictable with use of an appropriate random source used in combination with appropriate pseudo-random techniques. Compromising the security of the key generation method shall require at least as many operations as determining the value of the generated key.<br>Application note: For more details concerning this OSP, refer to [GP-CCCM]. |
| OSP.OPERATOR-KEYS | The security of the mobile operator keys (ISD keys) must be ensured by a well-defined security policy that covers generation, storage, distribution, destruction and recovery. This policy is enforced by the mobile operator in collaboration with the personalizer.<br>Application note: Token keys used to verify the tokens included in Delegated Management commands (that embed the signature of these commands) must be different for each (U)SIM card in usage. |

| | |
|---|---|
| OSP.KEY-GENERATION | The personalizer must enforce a policy ensuring that generated keys cannot be accessed in plaintext.<br>Application note: This can be applied by encrypting the generated key just after its generation with the public key of the recipient. |
| OSP.CASD-KEYS | The security domain keys of the CA must be securely generated and stored in the (U)SIM card during the personalization process. These keys are not modifiable after card issuance. |
| OSP.VASD-KEYS | The security domain keys of the VA must be securely generated and stored in the (U)SIM card during the personalization process. |
| OSP.KEY-CHANGE | The AP shall change its initial security domain keys (APSD) before any operation on its Security Domain. |
| OSP.SECURITY-DOMAINS | Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode. |
| OSP.QUOTAS | Security domains are subject to quotas of memory at creation. |
| OSP.PRODUCTION | Production and personalization environment must be trusted and secure as the TOE delivery occurs after Phase 6 of the TOE life cycle.<br>Application note: This OSP replaces A.PRODUCTION defined in PP (U)SIM. |
| OSP.PERSONALIZER | The personalizer under an Operator's Contract is in charge of the TOE personalization process before card issuance. He ensures the security of the keys he loads on the (U)SIM cards:<br>▪ Mobile operator keys including OTA keys (telecom keys either generated by the personalizer or by the mobile operator) and delegated management token keys<br>▪ Issuer Security Domain keys (ISD keys or Card issuer keys),<br>▪ Application Provider Security Domains keys (APSD keys).<br>▪ Controlling Authority Security Domain keys (CASD keys)<br>▪ Verification Authority Security Domain keys (VASD keys)<br>Application note: This OSP replaces A.PERSONALIZER defined in PP (U)SIM. |
| OSP.KEY-ESCROW | The key escrow is a trusted actor in charge of the secure storage of the initial AP keys generated by the TOE personalizer during initial personalization.<br>Application note: This OSP replaces A.KEY-ESCROW defined in PP (U)SIM. |

### *6.4.2. JCS Protection Profile*

According to [PP-USIM], the OSP listed in [PP-JCS] shall also be considered for the present evaluation.

| | |
|---|---|
| OSP.VERIFICATION | This policy shall ensure the consistency 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. See #.VERIFICATION for details. |

### *6.4.3. Supplementary OSPs*

The following OSPs shall also be considered for the present evaluation.

- OSP.SecureAPI, OSP.RNG and OSP.JCAPI-Services are related to additional services provided to applications by the TOE.
- OSP.TRUSTED-APPS-DEVELOPER and OSP.TRUSTED-APPS_PRE-ISSUANCE-LOADING are provided to manage pre-issuance.
- OSP.SERVICE_AUDIT and OSP.ACTIVATION-KEY-ESCROW are provided to manage service activation by OTA.

- OSP.Atomic_Activation, OSP.TOE_Identification and OSP.Additional_Code_Signing are provided to manage patch loading and activation.

| | |
|---|---|
| OSP.SecureAPI | The TOE must contribute to ensure that applications can optimize control on their sensitive operations. For that purpose, the TOE implements a dedicated API which provides security services to applications (e.g. secure array management, loss of data integrity detection, inconsistent execution flow detection, reaction against tearing or fault induction). |
| OSP.RNG | This policy shall ensure that the entropy of the random numbers provided by the TOE to applets using [JCAPI301] is sufficient. Thus attacker is not able to predict or obtain information on generated numbers. |
| OSP.JCAPI-Services | This policy shall ensure that hashing and checksum security services, defined in [JCAPI301] and provided by the TOE to applets, are secure. Thus attacker is not able to predict or obtain information on manipulated data. |
| OSP.TRUSTED-APPS-DEVELOPER | There are application developers (as Gemalto) considered as trusted by MNOs and Application Providers. The confidence in these actors has been obtained by audits of the development process and development environment performed by ITSEF during private scheme evaluations or Common Criteria composite evaluations.<br>As a consequence, the development process applied by a trusted developer provides confidence that applications developed by this actor are not aggressive versus the platform and other applications loaded on top of it. |
| OSP.TRUSTED-APPS_PRE-ISSUANCE-LOADING | For Pre-Issuance loading of trusted* applications, the process audited by ITSEF during private scheme evaluations or Common Criteria evaluations must be used.<br>* Application notes:<br>▪ An application is considered as trusted if it has been developed or verified by a trusted actor (as Gemalto).<br>▪ An application developed by a third party can be considered as a trusted application only if it has been verified and signed by verification authority. The application and associated signature will be verified by Gemalto prior authorizing loading in pre-issuance.<br>Therefore, the loading process applied by a trusted personalizer provides confidence that only trusted applications are loaded on the TOE, and consequently are not aggressive versus the platform and other applications loaded on top of it. |
| OSP.SERVICE_AUDIT | The MNO and GemActivate administrator (usually Gemalto) can audit optional platform service activation using remote service audit. |
| OSP.ACTIVATION-KEY-ESCROW | The key escrow is a trusted actor in charge of the secure storage of the activation keys generated and stored outside of the TOE and imported in the TOE by the TOE personalizer during initial personalization. He ensures the security of the keys for remote service activation. |
| OSP.Atomic_Activation | When the TOE supports ability to include additional code, additional code has to be loaded and installed on the initial TOE through an atomic activation to create the updated TOE.<br>Each additional code shall be identified with unique Identification Data. During such atomic activation, identification Data of the TOE have to be updated to clearly identified the updated TOE.<br>In case of interruption or incident during activation, the TOE shall |

| | |
|---|---|
| | remain in its initial state or fail secure. |
| OSP.TOE_Identification | Identification Data of the resulting updated TOE shall identify the Initial TOE and the activated additional code. Identification Data shall be protected in integrity. |
| OSP.Additional_Code_Signing | The additional code has to be signed with a cryptographic key according to relevant standard and generated certificate is associated to additional code.<br>Code signature must be checked during loading to assure authenticity and integrity of the loaded additional code and to assume that loading is authorized on the relevant TOE to the involved actor.<br>The cryptographic key shall be of sufficient quality and the process of key generation and certificate generation shall be appropriately secured to ensure (i) the confidentiality, authenticity and integrity of the key, (ii) the authenticity and integrity of certificate. |

## 6.5. SECURE USAGE ASSUMPTIONS

### *6.5.1. USIM Protection Profile*

The following assumptions are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| A.MOBILE-OPERATOR | The mobile operator is a trusted actor responsible for the mobile network and the associated OTA servers.<br>The mobile operator as Card issuer cannot get access or change the application data which belongs to the AP. |
| A.OTA-ADMIN | Administrators of the mobile operator OTA servers are trusted people. They are trained to use and administrate securely those servers. They have the means and the equipments to perform their tasks.<br>They are aware of the sensitivity of the assets they managed and the responsibilities associated to the administration of OTA servers.<br>Application note: OTA servers security guidance document with regular site inspections shall be employed to check the applicability of the rules. |
| A.APPS-PROVIDER | The AP is a trusted actor that provides basic or secure applications. He is responsible for his security domain keys (APSD keys).<br>Application note: An AP generally refers to the entity that issues the application. For instance it can be a financial institution for a payment application such as EMV or a transport operator for a transport application such as Calypso. |
| A.VERIFICATION-AUTHORITY | The VA is a trusted actor who is able to guarantee and check the digital signature attached to a basic or secure application.<br>Application note: As a consequence, it guarantees the success of the application validation or certification upon loading. |
| A.CONTROLLING-AUTHORITY | The CA is a trusted actor responsible for securing the APSD keys creation and personalization. He is responsible for his security domain keys (CASD keys). |

### *6.5.2. JCS Protection Profile*

The following assumptions from [PP-JCS] shall also be considered for the present evaluation.

| | |
|---|---|
| A.APPLET | Applets loaded post-issuance do not contain native methods. The Java Card specification explicitly "does not include support for native methods" ([JCVM22], §3.3) outside the API. |
| A.VERIFICATION | All the bytecodes are verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time. |

## 6.6. COMPOSITION TASKS – SECURITY PROBLEM DEFINITION PART

### *6.6.1. Statement of Compatibility – Threats part*

The following table (see next page) lists the relevant threats of the security target [ST_ST33G1M2], and provides the link to the threats on the composite-product, showing that there is no contradiction between the two.

Note: The additional threats defined in [ST_ST33G1M2] and related to DESFire are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security.

| IC relevant threat label | IC relevant threat title | IC relevant threat content | Link to the composite-product threats |
|---|---|---|---|
| T.Leak-Inherent | Inherent Information Leakage | An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets.<br>No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. | Analysis of the composite-product threats does not reveal any contradiction with this IC threat. |
| T.Phys-Probing | Physical Probing | An attacker may perform physical probing of the TOE in order<br>(i) to disclose User Data (ii) to disclose/reconstruct the Security IC Embedded Software or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. | T.PHYSICAL |
| T.Malfunction | Malfunction due to Environmental Stress | An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to<br>(i) modify security services of the TOE or<br>(ii) modify functions of the Security IC Embedded Software<br>(iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.<br>This may be achieved by operating the Security IC outside the normal operating conditions. | T.PHYSICAL |
| T.Phys-Manipulation | Physical Manipulation | An attacker may physically modify the Security IC in order to<br>(i) modify User Data<br>(ii) modify the Security IC Embedded Software<br>(iii) modify or deactivate security services of the TOE, or<br>(iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. | T.PHYSICAL |
| T.Leak-Forced | Forced Information Leakage | An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets even if the information leakage is not inherent but caused by the attacker. | Analysis of the composite-product threats does not reveal any contradiction with this IC threat. |
| T.Abuse-Func | Abuse of Functionality | An attacker may use functions of the TOE which may not be used after TOE Delivery in order to | Analysis of the composite-product threats does not reveal any |

| IC relevant threat label | IC relevant threat title | IC relevant threat content | Link to the composite-product threats |
|---|---|---|---|
| | | (i) disclose or manipulate User Data<br>(ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or<br>(iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or<br>(iv) enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software. | contradiction with this IC threat. |
| T.RND | Deficiency of Random Numbers | An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided. | Analysis of the composite-product threats does not reveal any contradiction with this IC threat.<br>Moreover, this IC threat participates to the enforcement of OSP.RNG. |
| T.Mem-Access | Memory Access Violation | Parts of the Security IC Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code). Any restrictions are defined by the security policy of the specific application context and must be implemented by the Security IC Embedded Software. | T.CONFID-APPLI-DATA<br>T.CONFID-JCS-DATA<br>T.INTEG-APPLI-DATA<br>T.INTEG-JCS-DATA<br>T.SID.1<br>T.SID.2<br>T.EXE-CODE.1 |

### 6.6.2. *Statement of compatibility – OSPs part*

The following table lists the relevant OSPs of the security target [ST_ST33G1M2], and provides the link to the OSPs related to the composite-product, showing that there is no contradiction between the two.

| IC OSP label | IC OSP content | Link to the composite product |
|---|---|---|
| P.Process-TOE | Protection during TOE Development and Production: an accurate identification is established for the TOE. This requires that each instantiation of the TOE carries this unique identification. | No contradiction with the present evaluation; the chip traceability information participates to the composite TOE identification. |
| P.Add-Functions | Additional Specific Security Functionality:<br>The TOE shall provide the following specific security functionality to the Security IC Embedded Software:<br>▪ Data Encryption Standard (DES)<br>▪ Triple Data Encryption Standard (3DES)<br>▪ Advanced Encryption Standard (AES) | The 3DES and AES hardware functionalities are used by the composite TOE.<br>The DES hardware functionality is not used by the composite TOE. |
| P.Controlled-ES-Loading | Controlled loading of the Security IC Embedded Software:<br>The TOE shall provide the capability to import the Security IC Embedded Software into the NVM, in a controlled manner, either before TOE delivery, under ST authority, either after TOE delivery, under the composite product manufacturer authority. | As mentioned in section 4.5, the UpTeq NFC3.2.2_Generic v1.0 software is loaded during phase 6 of the composite TOE life cycle (under Gemalto authority).<br>OSP.PRODUCTION states that the corresponding environment must be trusted and secure. |

Note: The additional OSPs defined in [ST_ST33G1M2] and related to DESFire/Neslib are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security, and Neslib is not present in the IC.

### 6.6.3. *Statement of compatibility – Assumptions part*

The following table (see next page) lists the relevant assumptions of the security target [ST_ST33G1M2], and provides the link to the assumptions related to the composite-product, showing that there is no contradiction between the two.

Note: The additional assumptions defined in [ST_ST33G1M2] and related to DESFire are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security.

| IC assumption label | IC assumption title | IC assumption content | IrPA | CfPA | SgPA | Link to the composite product |
|---|---|---|---|---|---|---|
| A.Process-Sec-IC | Protection during Packaging, Finishing and Personalization | It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use). This means that the Phases after TOE Delivery are assumed to be protected appropriately. | | X | X | • During phases 4, 5 and 6: CfPA<br>Fulfilled by the ALC composite-SARs and by the enforcement of OSP.PRODUCTION OSP.PERSONALIZER and OSP.KEY-ESCROW.<br>• During phase 7: SgPA<br>A.MOBILE-OPERATOR, A.OTA-ADMIN, A.APPS-PROVIDER, A.VERIFICATION-AUTHORITY, A.CONTROLLING-AUTHORITY and A.VERIFICATION. |
| A.Plat-Appl | Usage of Hardware Platform | The Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report. | | X | | Fulfilled by the composite-SAR ADV_COMP.1<br>(cf [CCDB], Appendix 1.2, §72 and §73) |
| A.Resp-Appl | Treatment of User Data | All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context. | | X | | O.KEY-MNGT<br>O.PIN-MNGT |

# 7. Security objectives

## 7.1. SECURITY OBJECTIVES FOR THE TOE

### *7.1.1. USIM Protection Profile*

The following TOE security objectives are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| O.CARD-MANAGEMENT | The TOE shall provide card management functionalities (loading, installation, extradition, deletion of applications and GP registry updates) in charge of the life cycle of the whole (U)SIM card and installed applications (applets).<br>The card manager, the application with specific rights responsible for the administration of the smart card, shall control the access to card management functions. It shall also implement the card issuer's policy on card management.<br><br>Application note: The card manager will be tightly connected in practice with the rest of the TOE, which in return shall very likely rely on the card manager for the effective enforcement of some of its security functions.<br>The mechanism used to ensure authentication of the TOE issuer, that manages the TOE, or of the Service Providers owning a Security Domain with card management privileges is a secure channel. This channel will be used afterwards to protect commands exchanged with the TOE in confidentiality and integrity.<br>The platform guarantees that only the ISD or the Service Providers owning a Security Domain with the appropriate privilege (Delegated Management) can manage the applications on the card associated with its Security Domain. This is done accordingly with the card issuer's policy on card management.<br>The actor performing the operation must beforehand authenticate with the Security Domain. In the case of Delegated Management, the card management command will be associated with an electronic signature (GlobalPlatform token) verified by the ISD before execution. |
| O.DOMAIN-RIGHTS | The Card issuer shall not get access or change personalized AP security domain keys which belong to the AP. Modification of a security domain keyset is restricted to the AP who owns the security domain.<br>Application note: APs have a set of keys that allows them to establish a secure channel between them and the platform. These keys sets are not known by the TOE issuer. The security domain initial keys are changed before any operation on the SD (OE.KEY-CHANGE) through standard PUT KEY procedures (if the initial keys were kept by key escrow) or through one of the SD personalization mechanisms described in Section 4.3.3 of [GP-UICC]. |
| O.APPLI-AUTH | The card manager shall enforce the application security policies established by the card issuer by requiring application authentication during application loading on the card.<br>Application note: Each application loaded onto the TOE has been signed by the VA. The VA will guarantee that the security policies established by the card issuer on applications are enforced. This authority is present on the TOE as a Security Domain whose role is to verify each signature at application loading.<br>The platform provides important extra features about application management and especially loading:<br>▪ Loaded applications are previously validated by an accredited laboratory for basic applications and certified by an accredited ITSEF for secure applications.<br>▪ All loaded applications are associated to a DAP signature generated by a VA which is verified at loading by the third party representative present on the platform (Mandated DAP verification). |
| O.COMM_AUTH | The TOE shall authenticate the origin of the card management requests that the card receives, and authenticate itself to the remote actor. |
| O.COMM_INTEGRITY | The TOE shall verify the integrity of the card management requests that the card |

| | |
|---|---|
| | receives. |
| O.COMM_CONFIDENTIALITY | The TOE shall be able to process card management requests containing encrypted data. |
| O.SCP-SUPPORT | The TOE OS shall support the following functionalities:<br>(1) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by the packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System.<br>(2) It provides secure low-level cryptographic processing to the Java Card System and GlobalPlatform.<br>(3) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism.<br>(4) It allows the Java Card System to store data in "persistent technology memory" or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection). |

### *7.1.2. JCS Protection Profile*

The following TOE security objectives from [PP-JCS] shall also be considered for the present evaluation.

| | |
|---|---|
| O.SID | The TOE shall uniquely identify every subject (applet, or package) before granting it access to any service. |
| O.FIREWALL | The TOE shall ensure controlled sharing of data containers owned by applets of different packages or the JCRE and between applets and the TSFs. See #.FIREWALL for details. |
| O.GLOBAL_ARRAYS_CONFID | The TOE shall ensure that the APDU buffer that is shared by all applications is always cleaned upon applet selection.<br>The TOE shall ensure that the global byte array used for the invocation of the install method of the selected applet is always cleaned after the return from the install method. |
| O.GLOBAL_ARRAYS_INTEG | The TOE shall ensure that only the currently selected applications may have a write access to the APDU buffer and the global byte array used for the invocation of the install method of the selected applet. |
| O.NATIVE | The only means that the Java Card VM shall provide for an application to execute native code is the invocation of a method of the Java Card API, or any additional API. See #.NATIVE for details. |
| O.OPERATE | The TOE must ensure continued correct operation of its security functions. See #.OPERATE for details. |
| O.REALLOCATION | The TOE shall ensure that the re-allocation of a memory block for the runtime areas of the Java Card VM does not disclose any information that was previously stored in that block. |
| O.RESOURCES | The TOE shall control the availability of resources for the applications. See #.RESOURCES for details. |
| O.ALARM | The TOE shall provide appropriate feedback information upon detection of a potential security violation. See #.ALARM for details. |
| O.CIPHER | The TOE shall provide a means to cipher sensitive data for applications in a secure way. In particular, the TOE must support cryptographic algorithms consistent with cryptographic usage policies and standards. See #.CIPHER for details. |
| O.KEY-MNGT | The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See #.KEYMNGT. |
| O.PIN-MNGT | The TOE shall provide a means to securely manage PIN objects. See #.PIN-MNGT |

| | |
|---|---|
| | for details.<br>Application note: 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 sensitive as that of the PIN. |
| O.TRANSACTION | The TOE must provide a means to execute a set of operations atomically. See #.TRANSACTION for details. |
| O.OBJ-DELETION | The TOE shall ensure the object deletion shall not break references to objects. See #.OBJ-DELETION for further details. |
| O.DELETION | The TOE shall ensure that both applet and package deletion perform as expected. See #.DELETION for details. |
| O.LOAD | The TOE shall ensure that the loading of a package into the card is safe.<br>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. |
| O.INSTALL | The TOE shall ensure that the installation of an applet performs as expected (See #.INSTALL for details). |
| O.SCP.IC | The SCP shall provide all IC security features against physical attacks. This security objective refers to the point (7) of the security aspect #.SCP: It is required that the IC is designed in accordance with a well-defined set of policies and Standards (likely specified in another protection profile), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of cryptographic keys. |
| O.SCP.RECOVERY | If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state.<br>This security objective refers to the security aspect #.SCP(1): The smart card platform must be secure with respect to the SFRs. Then after a power loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state. |

### *7.1.3. Supplementary TOE security objectives*

The following TOE security objectives shall also be considered for the present evaluation:

- O.Secure_API, O.RND and O.JCAPI-Services are related to additional services provided to applications by the TOE
- O.REMOTE_SERVICE_AUDIT and O.REMOTE_SERVICE_ACTIVATION are related to optional service activation by OTA.
- O.Secure_Load_ACode, O.Secure_AC_Activation and O.TOE_Identification are related to patch loading and activation.

| | |
|---|---|
| O.Secure_API | The TOE shall provide a dedicated API - named Secure API - to applications, so as to optimize control on their sensitive operations. The Secure API shall provide security services such as secure array management, loss of data integrity detection, inconsistent execution flow detection, reaction against tearing or fault induction. |
| O.RND | The TOE shall ensure that random numbers are not predictable and have sufficient entropy. |
| O.JCAPI-Services | The TOE shall ensure that data manipulated during SHA and CRC services as defined in [JCAPI301] cannot be observed. |
| O.REMOTE_SERVICE_AUDIT | The TOE shall perform remote service audit only when optional platform service audit is authorized and only by an authorized actor. Limited to [MNO or GemActivate Administrator (usually Gemalto)]. |
| O.REMOTE_SERVICE_ACTIVATION | The TOE shall perform remote optional platform service activation only when service activation is authorized and only by an authorized actor. Limited to [GemActivate Administrator (usually Gemalto)] under control of [MNO]. |
| O.Secure_Load_ACode | The TOE Loader shall check an evidence of authenticity and integrity of the Additional Code to be loaded.<br>The TOE Loader enforces that only an allowed version of the Additional Code can be loaded. The TOE Loader shall forbid the loading of an Additional Code not intended to be assembled with the TOE.<br>During the Load phase of the Additional Code, the TOE shall remain secure. |
| O.Secure_AC_Activation | Activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the updated TOE shall be completed before activation.<br>If the atomic activation is successful, then the resulting product is the updated TOE, otherwise (in case of interruption or incident which prevents the forming of the updated TOE), the TOE shall remain in its initial state or fail securely. |
| O.TOE_Identification | The Identification Data identifies the TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data.<br>After Atomic Activation of the Additional Code, the Identification Data of the updated TOE allows identifications of both the Initial TOE and Additional Code.<br>The user must be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the updated TOE. |

## 7.2. SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

### *7.2.1. USIM Protection Profile*

The following security objectives for the operational environment are listed in [PP-USIM] and shall be considered for the present evaluation.

| | |
|---|---|
| OE.MOBILE-OPERATOR | The mobile operator shall be a trusted actor responsible for the mobile network and the associated OTA servers. |
| OE.OTA-ADMIN | Administrators of the mobile operator OTA servers shall be trusted people. They shall be trained to use and administrate those servers. They have the means and the equipments to perform their tasks. They must be aware of the sensitivity of the assets they manage and the responsibilities associated to the administration of OTA servers.<br><br>Application note: One possible realisation of this assumption is the enforcement of security rules defined in an OTA servers security guidance document with regular site inspections to check the applicability of the rules. |
| OE.APPS-PROVIDER | The AP shall be a trusted actor that provides basic or secure application. He must be responsible of his security domain keys. |

| | |
|---|---|
| OE.VERIFICATION-AUTHORITY | The VA should be a trusted actor who is able to guarantee and check the digital signature attached to an application. |
| OE.KEY-ESCROW | The key escrow shall be a trusted actor in charge of the secure storage of the AP initial keys generated by the personalizer. |
| OE.PERSONALIZER | The personalizer shall be a trusted actor in charge of the personalization process. He must ensure the security of the keys it manages and loads into the card:<br>▪ Mobile operator keys including OTA keys (telecom keys either generated by the personalizer or by the mobile operator),<br>▪ Issuer Security Domain keys (ISD keys),<br>▪ Application Provider Security Domain keys (APSD keys).<br>▪ Controlling Authority Security Domain keys (CASD keys) |
| OE.CONTROLLING-AUTHORITY | The CA shall be a trusted actor responsible for securing the APSD keys creation and personalisation. He must be responsible for his security domain keys (CASD keys). |
| OE.PRODUCTION | Production and personalization environment if the TOE delivery occurs before Phase 6 of the TOE life cycle must be trusted and secure. |
| OE.SECURE-APPS-CERTIFICATION | Secure applications must be evaluated and certified at a security level higher or equal than the one of the current Protection Profile. |
| OE.BASIC-APPS-VALIDATION | Basic applications must be analysed during the validation process in order to ensure that the rules for correct usage of the TOE are still enforced. |
| OE.AID-MANAGEMENT | The VA or the MNO shall verify that the AID of the application being loaded does not impersonate the AID known by another application on the card for the use of shareable services. |
| OE.OTA-LOADING | Application code, validated or certified depending on the application, is loaded "Over The Air" (OTA) onto (U)SIM Platform using OTA servers. This process should protect the confidentiality and the integrity of the loaded application code. |
| OE.OTA-SERVERS | The mobile operator must enforce a policy to ensure the security of the applications stored on its servers. |
| OE.AP-KEYS | The SD keys personalizer, the AP and the key escrow must enforce a security policy on SD keys in order to secure their transmission. |
| OE.OPERATOR-KEYS | The security of the mobile operator keys must be ensured in the environment of the TOE. |
| OE.KEY-GENERATION | The personalizer must ensure that the generated keys cannot be accessed by unauthorized users. |
| OE.CA-KEYS | The security domain keys of the CA must be securely generated prior storage in the (U)SIM card. |
| OE.VA-KEYS | The security domain keys of the VA must be securely generated prior storage in the (U)SIM card. |
| OE.KEY-CHANGE | The AP must change its security domain initial keys before any operation on it. |
| OE.SECURITY-DOMAINS | Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode. |
| OE.QUOTAS | Security domains are subject to quotas of memory at creation. |
| OE.SHARE-CONTROL | All applications (basic and secure applications) must have means to identify the applications with whom they share data using the Shareable Interface.<br>Application note: If an application implementing a Shareable Interface has to share data with a new application, it has to be updated, and thus re-validated, to take into account the identification of this new application (through its AID for instance) before sharing data. |

### *7.2.2. JCS Protection Profile*

The following security objectives for the operational environment are listed in [PP-JCS] and shall also be considered for the present evaluation.

| | |
|---|---|
| OE.APPLET | No applet loaded post-issuance shall contain native methods. |
| OE.VERIFICATION | All the bytecodes shall be verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time. See #.VERIFICATION |

| | |
|---|---|
| | for details.<br>Additionally, the applet shall follow all the recommendations, if any, mandated in the platform guidance for maintaining the isolation property of the platform.<br>*Application note:* Constraints to maintain the isolation property of the platform are provided by the platform developer in application development guidance. The constraints apply to all application code loaded in the platform. |

### *7.2.3. Supplementary security objectives for the operational environment*

The following security objectives for the operational environment shall also be considered for the present evaluation:

| | |
|---|---|
| OE.CODE-EVIDENCE | For application code loaded pre-issuance, evaluated technical measures implemented by the TOE or audited organizational measures must ensure that loaded application has not been changed since the code verifications required in OE.VERIFICATION.<br>For application code loaded post-issuance and verified off-card according to the requirements of OE.VERIFICATION, the verification authority shall provide digital evidence to the TOE that the application code has not been modified after the code verification and that he is the actor who performed code verification.<br>For application code loaded post-issuance and partially or entirely verified on-card, technical measures must ensure that the verification required in OE.VERIFICATION is performed. On-card bytecode verifier is out of the scope of the present evaluation.<br>*Application note:* For application code loaded post-issuance and verified off-card, the integrity and authenticity evidence can be achieved by electronic signature of the application code, after code verification, by the actor who performed the verification. |
| OE.TRUSTED-APPS-DEVELOPER | The trusted application developer shall be a trusted actor that provides basic or secure application where correct usage of the TOE has been verified applying a secure development process in a secure development environment. |
| OE.TRUSTED-APPS_PRE-ISSUANCE-LOADING | The trusted pre-issuance loading on the platform must be done only using verified applets, and applying an audited process in a secure environment. |
| OE.GEMACTIVATE-ADMIN | The GemActivate administrator shall be a trusted actor responsible for patch loading/activation and optional platform service activation in post issuance. Activation is under the control of the mobile operator as activation is done using OTA communication with MNO OTA servers and associated keys stored in the TOE. |
| OE.ACTIVATION-KEY-ESCROW | The key escrow is a trusted actor that must ensure the security of the keys used for patch loading and remote service activation during generation, storage, importation in TOE and usage. |
| OE.Secure_ACode_Management | The process associated to key and certificate management linked to additional code shall take place in a secure and audited environment.<br>The key generation process shall guarantee that cryptographic keys shall be of sufficient quality and appropriately secured to ensure confidentiality, authenticity and integrity of the keys.<br>The certificate generation process shall guarantee the authenticity and integrity of the certificate. |

## 7.3. SECURITY OBJECTIVES RATIONALE

### *7.3.1. Threats, OSPs and Assumptions coverage – Mapping tables*

| | T.PHYSICAL | T.INTEG-USER-DATA | T.COM_EXPLOIT | T.UNAUTHORIZED_CARD_MNGT | T.LIFE_CYCLE | T.UNAUTHORIZED_ACCESS | T.CONFID-APPLI-DATA | T.CONFID-JCS-DATA | T.CONFID-JCS-CODE | T.INTEG-APPLI-CODE | T.INTEG-APPLI-CODE.LOAD | T.INTEG-APPLI-DATA | T.INTEG-APPLI-DATA.LOAD | T.INTEG-JCS-DATA | T.INTEG-JCS-CODE | T.SID.1 | T.SID.2 | T.EXE-CODE.1 | T.EXE-CODE.2 | T.NATIVE | T.RESOURCES | T.DELETION | T.INSTALL | T.OBJ-DELETION | T.UNAUTHORIZED_ACCESS_TO_ SERVICE | T.UNAUTHORIZED_TOE_CODE_UPDATE | T.UNAUTHORIZED_TOE_CODE_UPDATE_BLOCK | T.TOE_IMPERSONATION | T.TOE_CLONING |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| O.CARD-MANAGEMENT | | | | X | X | | X | X | X | X | X | X | X | X | X | X | | | | | | X | X | | | | | | |
| O.DOMAIN-RIGHTS | | X | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| O.APPLI-AUTH | | | | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_AUTH | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_INTEGRITY | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_CONFIDENTIALITY | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SCP-SUPPORT | X | X | | | | | X | X | | | | X | | X | | | X | | | | X | | | | | | | | |
| O.SID | | | | | | | X | X | | | | X | | X | | X | X | | | | | | | | | | | | |
| O.FIREWALL | | | | | | | X | X | | | | X | | X | | X | X | X | | | | | | | | | | | |
| O.GLOBAL_ARRAYS_CONFID | | | | | | | X | | | | | | | | | X | | | | | | | | | | | | | |
| O.GLOBAL_ARRAYS_INTEG | | | | | | | | | | | | X | | | | X | | | | | | | | | | | | | |
| O.NATIVE | | | | | | | | | X | X | | | | | X | | | | | X | | | | | | | | | |
| O.OPERATE | | | | | | | X | X | | | | X | | X | | | X | | | | X | | | | | | | | |
| O.REALLOCATION | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | |
| O.RESOURCES | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | |
| O.ALARM | | | | | | | X | X | | | | X | | X | | | | | | | | | | | | | | | |
| O.CIPHER | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | |
| O.KEY-MNGT | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | |
| O.PIN-MNGT | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | |
| O.TRANSACTION | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | |
| O.OBJ-DELETION | | | | | | | | | | | | | | | | | | | | | | | | X | | | | | |
| O.DELETION | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | |
| O.LOAD | | | | | | | | | | | X | | X | | | | | | | | | | X | | | | | | |
| O.INSTALL | | | | | | | | | | | | | | | | X | X | | | | X | | X | | | | | | |
| O.SCP.IC | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SCP.RECOVERY | | | | | | | X | X | | | | X | | X | | | X | | | | X | | | | | | | | |
| O.Secure_API | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.RND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.JCAPI-Services | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.REMOTE_SERVICE_AUDIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.REMOTE_SERVICE_ACTIVATION | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| O.Secure_Load_ACode | | | | | | | | | | | | | | | | | | | | | | | | | | X | | X | |
| O.Secure_AC_Activation | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | | |
| O.TOE_Identification | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X |
| OE.MOBILE-OPERATOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.OTA-ADMIN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.APPS-PROVIDER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OE.VERIFICATION-AUTHORITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.KEY-ESCROW | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.PERSONALIZER | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.CONTROLLING-AUTHORITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.PRODUCTION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.SECURE-APPS-CERTIFICATION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.BASIC-APPS-VALIDATION | | | | | | | X | | | X | | X | | X | | | | X | X | X | | | | | | | | |
| OE.AID-MANAGEMENT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.OTA-LOADING | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.OTA-SERVERS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.AP-KEYS | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.OPERATOR-KEYS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.KEY-GENERATION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.CA-KEYS | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.VA-KEYS | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.KEY-CHANGE | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.SECURITY-DOMAINS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.QUOTAS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.SHARE-CONTROL | | | | | | X | | | | | | | | | | | | | | | | | | | | | | |
| OE.APPLET | | | | | | | | | | | | | | | | | | | | X | | | | | | | | |
| OE.VERIFICATION | | | | | | | X | X | X | X | | X | | X | X | | | X | X | X | | | | | | | | |
| OE.CODE-EVIDENCE | | | | | | | | | | X | X | X | X | X | | | | | | | | | | | | | | |
| OE.TRUSTED-APPS-DEVELOPER | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.TRUSTED-APPS_PRE-ISSUANCE-LOADING | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.GEMACTIVATE-ADMIN | | | | | | | | | | | | | | | | | | | | | | | | | X | | | |
| OE.ACTIVATION-KEY-ESCROW | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.Secure_ACode_Management | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | | X |

*Table 2: Threats coverage by security objectives – Mapping table*

| | OSP.SECURE-APPS-CERTIFICATION | OSP.BASIC-APPS-VALIDATION | OSP.SHARE-CONTROL | OSP.AID-MANAGEMENT | OSP.OTA-LOADING | OSP.OTA-SERVERS | OSP.APSD-KEYS | OSP.OPERATOR-KEYS | OSP.KEY-GENERATION | OSP.CASD-KEYS | OSP.VASD-KEYS | OSP.KEY-CHANGE | OSP.SECURITY-DOMAINS | OSP.QUOTAS | OSP.PRODUCTION | OSP.PERSONALIZER | OSP.KEY-ESCROW | OSP.VERIFICATION | OSP.SecureAPI | OSP.RNG | OSP.JCAPI-Services | OSP.TRUSTED-APPS-DEVELOPER | OSP.TRUSTED-APPS_PRE-ISSUANCE-LOADING | OSP.SERVICE_AUDIT | OSP.ACTIVATION-KEY-ESCROW | OSP.Atomic_Activation | OSP.TOE_Identification | OSP.Additional_Code_Signing |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| O.CARD-MANAGEMENT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.DOMAIN-RIGHTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.APPLI-AUTH | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_AUTH | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_INTEGRITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.COMM_CONFIDENTIALITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SCP-SUPPORT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SID | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.FIREWALL | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.GLOBAL_ARRAYS_CONFID | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.GLOBAL_ARRAYS_INTEG | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.NATIVE | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.OPERATE | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.REALLOCATION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.RESOURCES | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.ALARM | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.CIPHER | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.KEY-MNGT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.PIN-MNGT | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.TRANSACTION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.OBJ-DELETION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.DELETION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.LOAD | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.INSTALL | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SCP.IC | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.SCP.RECOVERY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.Secure_API | | | | | | | | | | | | | | | | | | | X | | | | | | | | | |
| O.RND | | | | | | | | | | | | | | | | | | | | X | | | | | | | | |
| O.JCAPI-Services | | | | | | | | | | | | | | | | | | | | | X | | | | | | | |
| O.REMOTE_SERVICE_AUDIT | | | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| O.REMOTE_SERVICE_ACTIVATION | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.Secure_Load_ACode | | | | | | | | | | | | | | | | | | | | | | | | | | X | | X |
| O.Secure_AC_Activation | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| O.TOE_Identification | | | | | | | | | | | | | | | | | | | | | | | | | | | X | |
| OE.MOBILE-OPERATOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.OTA-ADMIN | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.APPS-PROVIDER | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.VERIFICATION-AUTHORITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.KEY-ESCROW | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| OE.PERSONALIZER | | | | | | | | | | | | | | | | X | | | | | | | | | | | | |
| OE.CONTROLLING-AUTHORITY | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.PRODUCTION | | | | | | | | | | | | | | | X | | | | | | | | | | | | | |
| OE.SECURE-APPS-CERTIFICATION | X | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OE.BASIC-APPS-VALIDATION | | X | | | | | | | | | | | | | | | | X | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **OE.AID-MANAGEMENT** | | | | X | | | | | | | | | | | | | | | | | | | | | | | |
| **OE.OTA-LOADING** | | | | | X | | | | | | | | | | | | | | | | | | | | | | |
| **OE.OTA-SERVERS** | | | | | | X | | | | | | | | | | | | | | | | | | | | | |
| **OE.AP-KEYS** | | | | | | | X | | | | | | | | | | | | | | | | | | | | |
| **OE.OPERATOR-KEYS** | | | | | | | | X | | | | | | | | | | | | | | | | | | | |
| **OE.KEY-GENERATION** | | | | | | | | | X | | | | | | | | | | | | | | | | | | |
| **OE.CA-KEYS** | | | | | | | | | | X | | | | | | | | | | | | | | | | | |
| **OE.VA-KEYS** | | | | | | | | | | | X | | | | | | | | | | | | | | | | |
| **OE.KEY-CHANGE** | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| **OE.SECURITY-DOMAINS** | | | | | | | | | | | | | X | | | | | | | | | | | | | | |
| **OE.QUOTAS** | | | | | | | | | | | | | | X | | | | | | | | | | | | | |
| **OE.SHARE-CONTROL** | | | X | | | | | | | | | | | | | | | | | | | | | | | | |
| **OE.APPLET** | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **OE.VERIFICATION** | | | | | | | | | | | | | | | | | X | | | | | | | | | | |
| **OE.CODE-EVIDENCE** | | | | | | | | | | | | | | | | | X | | | | | | | | | | |
| **OE.TRUSTED-APPS-DEVELOPER** | | | | | | | | | | | | | | | | | | | | X | | | | | | | |
| **OE.TRUSTED-APPS_PRE-ISSUANCE-LOADING** | | | | | | | | | | | | | | | | | | | | | X | | | | | | |
| **OE.GEMACTIVATE-ADMIN** | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **OE.ACTIVATION-KEY-ESCROW** | | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| **OE.Secure_ACode_Management** | | | | | | | | | | | | | | | | | | | | | | | | | | | X |

***Table 3: OSP coverage by security objectives – Mapping table***

| | A.MOBILE-OPERATOR | A.OTA-ADMIN | A.APPS-PROVIDER | A.VERIFICATION-AUTHORITY | A.CONTROLLING-AUTHORITY | A.APPLET | A.VERIFICATION |
|---|---|---|---|---|---|---|---|
| O.CARD-MANAGEMENT | | | | | | | |
| O.DOMAIN-RIGHTS | | | | | | | |
| O.APPLI-AUTH | | | | | | | |
| O.COMM_AUTH | | | | | | | |
| O.COMM_INTEGRITY | | | | | | | |
| O.COMM_CONFIDENTIALITY | | | | | | | |
| O.SCP-SUPPORT | | | | | | | |
| O.SID | | | | | | | |
| O.FIREWALL | | | | | | | |
| O.GLOBAL_ARRAYS_CONFID | | | | | | | |
| O.GLOBAL_ARRAYS_INTEG | | | | | | | |
| O.NATIVE | | | | | | | |
| O.OPERATE | | | | | | | |
| O.REALLOCATION | | | | | | | |
| O.RESOURCES | | | | | | | |
| O.ALARM | | | | | | | |
| O.CIPHER | | | | | | | |
| O.KEY-MNGT | | | | | | | |
| O.PIN-MNGT | | | | | | | |
| O.TRANSACTION | | | | | | | |
| O.OBJ-DELETION | | | | | | | |
| O.DELETION | | | | | | | |
| O.LOAD | | | | | | | |
| O.INSTALL | | | | | | | |
| O.SCP.IC | | | | | | | |
| O.SCP.RECOVERY | | | | | | | |
| O.Secure_API | | | | | | | |
| O.RND | | | | | | | |
| O.JCAPI-Services | | | | | | | |
| O.REMOTE_SERVICE_AUDIT | | | | | | | |
| O.REMOTE_SERVICE_ACTIVATION | | | | | | | |
| O.Secure_Load_ACode | | | | | | | |
| O.Secure_AC_Activation | | | | | | | |
| O.TOE_Identification | | | | | | | |
| OE.MOBILE-OPERATOR | X | | | | | | |
| OE.OTA-ADMIN | | X | | | | | |
| OE.APPS-PROVIDER | | | X | | | | |
| OE.VERIFICATION-AUTHORITY | | | | X | | | |
| OE.KEY-ESCROW | | | | | | | |
| OE.PERSONALIZER | | | | | | | |
| OE.CONTROLLING-AUTHORITY | | | | | X | | |
| OE.PRODUCTION | | | | | | | |
| OE.SECURE-APPS-CERTIFICATION | | | | | | | |
| OE.BASIC-APPS-VALIDATION | | | | | | | |
| OE.AID-MANAGEMENT | | | | | | | |
| OE.OTA-LOADING | | | | | | | |
| OE.OTA-SERVERS | | | | | | | |
| OE.AP-KEYS | | | | | | | |
| OE.OPERATOR-KEYS | | | | | | | |
| OE.KEY-GENERATION | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| OE.CA-KEYS | | | | | | | |
| OE.VA-KEYS | | | | | | | |
| OE.KEY-CHANGE | | | | | | | |
| OE.SECURITY-DOMAINS | | | | | | | |
| OE.QUOTAS | | | | | | | |
| OE.SHARE-CONTROL | | | | | | | |
| OE.APPLET | | | | | | X | |
| OE.VERIFICATION | | | | | | | X |
| OE.CODE-EVIDENCE | | | | | | | X |
| OE.TRUSTED-APPS-DEVELOPER | | | | | | | |
| OE.TRUSTED-APPS_PRE-ISSUANCE-LOADING | | | | | | | |
| OE.GEMACTIVATE-ADMIN | | | | | | | |
| OE.ACTIVATION-KEY-ESCROW | | | | | | | |
| OE.Secure_ACode_Management | | | | | | | |

*Table 4: Assumptions coverage by security objectives – Mapping table*

### *7.3.2. Threats coverage – Rationale*

**T.PHYSICAL**
This threat is countered by physical protections which rely on the underlying platform. The security objectives O.SCP-SUPPORT and O.SCP.IC protect sensitive assets of the platform against loss of integrity and confidentiality and especially ensure the TSFs cannot be bypassed or altered.

**T.INTEG-USER-DATA**
- The security objective O.SCP-SUPPORT provides functionality to ensure atomicity of sensitive operations, secure low level access control and protection against bypassing of the security features of the TOE. In particular, it explicitly ensures the independent protection in integrity of the platform data.
- The security objectives O.DOMAIN-RIGHTS, OE.CA-KEYS, OE.VA-KEYS and OE.AP-KEYS ensure that personalization of the application by its associated security domain is only performed by the authorized AP.
- The security objectives from [PP-JCS] covering the threat T.INTEG-APPLI-DATA also cover this threat.

**T.COM_EXPLOIT**
This threat is covered by the following security objectives:
- O.COMM_AUTH prevents unauthorized users from initiating a malicious card management operation.
- O.COMM_INTEGRITY protects the integrity of the card management data while it is in transit to the (U)SIM card.
- O.COMM_CONFIDENTIALITY prevents from disclosing encrypted data transiting to the (U)SIM card.

**T.UNAUTHORIZED_CARD_MNGT**
This threat is covered by the following security objectives:
- O.CARD-MANAGEMENT controls the access to card management functions such as the loading, installation, extradition or deletion of applets.
- O.COMM_AUTH prevents unauthorized users from initiating a malicious card management operation.
- O.COMM_INTEGRITY protects the integrity of the card management data while it is in transit to the (U)SIM card.
- O.APPLI-AUTH which requires for loading all applications to be authenticated.
- O.DOMAIN-RIGHTS which restricts the modification of an AP security domain keyset to the AP who owns it.

**T.LIFE_CYCLE**
This threat is covered by the security objectives:
- O.CARD-MANAGEMENT that controls the access to card management functions such as the loading, installation, extradition or deletion of applets and prevent attacks intended to modify or exploit the current life cycle of applications
- O.DOMAIN-RIGHTS that restricts the use of an AP security domain keysets, and thus the management of the applications related to this SD, to the AP who owns it.

**T.UNAUTHORIZED_ACCESS**
This threat is covered by the security objective on the operational environment of the TOE OE.SHARE-CONTROL which ensures that sharing objects functionality is strictly controlled to stop data transitive flows between applets and thus stop access to unauthorized data.

**T.CONFID-APPLI-DATA**
- This threat is countered by the security objective for the operational environment regarding bytecode verification OE.VERIFICATION and guide application OE.BASIC-APPS-VALIDATION. It is also covered by the isolation commitments stated in the O.FIREWALL

objective. It relies in its turn on the correct identification of applets stated in O.SID. Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the O.OPERATE objective.

- As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.
- The objective O.CARD-MANAGEMENT contributes to cover this threat by controlling the access to card management functions.
- The objectives O.SCP.RECOVERY and O.SCP-SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.
- As applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys, PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) shall contribute in covering this threat by controlling the sharing of the global PIN between the applets.
- Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The disclosure of such data is prevented by the security objective O.GLOBAL_ARRAYS_CONFID.
- Finally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the O.REALLOCATION objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused.

**T.CONFID-JCS-DATA**

- This threat is covered by bytecode verification OE.VERIFICATION and the isolation commitments stated in the O.FIREWALL security objective. This latter objective also relies in its turn on the correct identification of applets stated in O.SID.
- Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the O.OPERATE objective.
- As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.
- The objective O.CARD-MANAGEMENT contributes to cover this threat by controlling the access to card management functions.
- The objectives O.SCP.RECOVERY and O.SCP-SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

**T.CONFID-JCS-CODE**

- This threat is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of those instructions enables reading a piece of code, no Java Card applet can therefore be executed to disclose a piece of code.
- Native applications are also harmless because of the objective O.NATIVE, so no application can be run to disclose a piece of code.
- The (#.VERIFICATION) security aspect is addressed in this PP by the objective for the environment OE.VERIFICATION.
- The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

**T.INTEG-APPLI-CODE**

- This threat is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION.
- Native applications are also harmless because of the objectives O.NATIVE and OE.BASIC-APPS-VALIDATION, so no application can be run to modify a piece of code.
- The objective O.CARD-MANAGEMENT contributes to cover this threat by controlling the access to card management functions.
- The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.

**T.INTEG-APPLI-CODE.LOAD**

- This threat is countered by the security objective O.LOAD which ensures that the loading of packages is done securely and thus preserves the integrity of packages code.
- The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
- By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

**T.INTEG-APPLI-DATA**

- This threat is countered by bytecode verification (OE.VERIFICATION and OE.BASIC-APPS-VALIDATION) and the isolation commitments stated in the O.FIREWALL objective. This latter objective also relies in its turn on the correct identification of applets stated in O.SID.
- Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the O.OPERATE objective.
- As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.
- The objective O.CARD-MANAGEMENT contributes to cover this threat by controlling the access to card management functions.
- The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
- The objectives O.SCP.RECOVERY and O.SCP-SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.
- Concerning the confidentiality and integrity of application sensitive data, as applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys and PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) is also concerned.
- Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The integrity of the information stored in that buffer is ensured by the objective O.GLOBAL_ARRAYS_INTEG.
- Finally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the O.REALLOCATION objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused.

**T.INTEG-APPLI-DATA.LOAD**

- This threat is countered by the security objective O.LOAD which ensures that the loading of packages is done securely and thus preserves the integrity of applications data.

- The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
- By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

**T.INTEG-JCS-DATA**

- This threat is countered by bytecode verification (OE.VERIFICATION) and guidance verification (OE.BASIC-APPS-VALIDATION) and the isolation commitments stated in the O.FIREWALL objective. This latter objective also relies in its turn on the correct identification of applets stated in O.SID.
- Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the O.OPERATE objective.
- As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.
- The objective O.CARD-MANAGEMENT contributes to cover this threat by controlling the access to card management functions.
- The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
- The objectives O.SCP.RECOVERY and O.SCP-SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

**T.INTEG-JCS-CODE**

- This threat is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can be run to modify a piece of code.
- The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION.
- The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

**T.SID.1**

- As impersonation is usually the result of successfully disclosing and modifying some assets, this threat is mainly countered by the objectives concerning the isolation of application data (like PINs), ensured by the (O.FIREWALL). Uniqueness of subject-identity (O.SID) also participates to face this threat. It should be noticed that the AIDs, which are used for applet identification, are TSF data.
- In this configuration, usurpation of identity resulting from a malicious installation of an applet on the card is covered by the objective O.INSTALL.
- The installation parameters of an applet (like its name) are loaded into a global array that is also shared by all the applications. The disclosure of those parameters (which could be used to impersonate the applet) is countered by the objectives O.GLOBAL_ARRAYS_CONFID and O.GLOBAL_ARRAYS_INTEG.
- The objective O.CARD-MANAGEMENT contributes, by preventing usurpation of identity resulting from a malicious installation of an applet on the card, to counter this threat.

**T.SID.2**

- This is covered by integrity of TSF data, subject-identification (O.SID), the firewall (O.FIREWALL) and its good working order (O.OPERATE).
- The objective O.INSTALL contributes to counter this threat by ensuring that installing an applet has no effect on the state of other applets and thus can't change the TOE's attribution of privileged roles.

- The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE objective of the TOE, so they are indirectly related to the threats that this latter objective contributes to counter.

**T.EXE-CODE.1**

- Unauthorized execution of a method is prevented by the objectives OE.VERIFICATION and OE.BASIC-APPS-VALIDATION. This threat particularly concerns the point (8) of the security aspect #VERIFICATION (access modifiers and scope of accessibility for classes, fields and methods).
- The O.FIREWALL objective is also concerned, because it prevents the execution of non-shareable methods of a class instance by any subject apart from the class instance owner.

**T.EXE-CODE.2**

Unauthorized execution of a method fragment or arbitrary data is prevented by the objectives OE.VERIFICATION and OE.BASIC-APPS-VALIDATION. This threat particularly concerns those points of the security aspect related to control flow confinement and the validity of the method references used in the bytecodes.

**T.NATIVE**

- This threat is countered by O.NATIVE which ensures that a Java Card applet can only access native methods indirectly that is, through an API which is assumed to be secure thanks to OE.BASIC-APPS-VALIDATION.
- OE.APPLET also covers this threat by ensuring that no native applets shall be loaded in post-issuance.
- In addition to this, the bytecode verifier also prevents the program counter of an applet to jump into a piece of native code by confining the control flow to the currently executed methods (OE.VERIFICATION and OE.BASIC-APPS-VALIDATION).

**T.RESOURCES**

- This threat is directly countered by objectives on resource-management (O.RESOURCES) for runtime purposes and good working order (O.OPERATE) in a general manner.
- Consumption of resources during installation and other card management operations are covered, in case of failure, by O.INSTALL.
- Finally, the objectives O.SCP.RECOVERY and O.SCP-SUPPORT are intended to support the O.OPERATE and O.RESOURCES objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

**T.DELETION**

- This threat is covered by the O.DELETION security objective which ensures that both applet and package deletion perform as expected.
- The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

**T.INSTALL**

- This threat is covered by the security objective O.INSTALL which ensures that the installation of an applet performs as expected and the security objectives O.LOAD which ensures that the loading of a package into the card is safe.
- The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

**T.OBJ-DELETION**

This threat is covered by the O.OBJ-DELETION security objective which ensures that object deletion shall not break references to objects.

**T.UNAUTHORIZED_ACCESS_TO_SERVICE**

This threat is countered by the security objectives O.REMOTE_SERVICE_ACTIVATION and OE.GEMACTIVATE-ADMIN where only an authorized and trusted actor is able to activate optional services.

**T.UNAUTHORIZED_TOE_CODE_UPDATE**
This threat is countered by the security objectives O.Secure_Load_ACode, O.Secure_AC_Activation and OE.Secure_ACode_Management.

**T.UNAUTHORIZED_TOE_CODE_UPDATE_BLOCK**
This threat is countered by the security objectives O.Secure_AC_Activation and OE.Secure_ACode_Management.

**T.TOE_IMPERSONATION**
This threat is countered by the security objectives O.Secure_Load_ACode and O.TOE_Identification.

**T.TOE_CLONING**
This threat is countered by the security objectives O.TOE_Identification and OE.Secure_ACode_Management.

### *7.3.3. OSP coverage – Rationale*

**OSP.SECURE-APPS-CERTIFICATION**
This OSP is enforced by the security objective for the operational environment of the TOE OE.SECURE-APPS-CERTIFICATION.

**OSP.BASIC-APPS-VALIDATION**
This OSP is enforced by the security objective for the operational environment of the TOE OE.BASIC-APPS-VALIDATION.

**OSP.SHARE-CONTROL**
This OSP is directly enforced by the security objective for the operational environment of the TOE OE.SHARE-CONTROL.

**OSP.AID-MANAGEMENT**
This OSP is directly enforced by the security objective for the operational environment of the TOE OE.AID-MANAGEMENT.

**OSP.OTA-LOADING**
This OSP is enforced by the security objective for the operational environment of the TOE OE.OTA-LOADING.

**OSP.OTA-SERVERS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.OTA-SERVERS.

**OSP.APSD-KEYS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.AP KEYS.

**OSP.OPERATOR-KEYS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.OPERATOR-KEYS.

**OSP.KEY-GENERATION**
This OSP is enforced by the security objective for the operational environment of the TOE OE.KEY-GENERATION.

**OSP.CASD-KEYS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.CA KEYS.

**OSP.VASD-KEYS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.VA KEYS.

**OSP.KEY-CHANGE**
This OSP is enforced by the security objective for the operational environment of the TOE OE.KEY CHANGE.

**OSP.SECURITY-DOMAINS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.SECURITY-DOMAINS.

**OSP.QUOTAS**
This OSP is enforced by the security objective for the operational environment of the TOE OE.QUOTAS.

**OSP.PRODUCTION**
This OSP is directly upheld by OE.PRODUCTION.

**OSP.PERSONALIZER**
This OSP is directly upheld by OE.PERSONALIZER.

**OSP.KEY-ESCROW**
This Security policy is directly upheld by OE.KEY-ESCROW.

**OSP.VERIFICATION**
- This policy is upheld by the security objectives of the environment OE.VERIFICATION and OE.BASIC-APPS-VALIDATION which guarantees that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time.
- This policy is also upheld by the security objective of the environment OE.CODE-EVIDENCE which ensures that evidences exist that the application code has been verified and not changed after verification.

**OSP.SecureAPI**
This OSP is enforced by the TOE security objective O.Secure_API.

**OSP.RNG**
This OSP is enforced by the TOE security objective O.RND.

**OSP.JCAPI-Services**
This OSP is enforced by the TOE security objective O.JCAPI-Services.

**OSP.TRUSTED-APPS-DEVELOPER**
This OSP is enforced by the security objective OE.TRUSTED-APPS-DEVELOPER.

**OSP.TRUSTED-APPS_PRE-ISSUANCE-LOADING**
This OSP is enforced by the security objective OE.TRUSTED-APPS_PRE-ISSUANCE-LOADING.

**OSP.SERVICE_AUDIT**
This OSP is directly enforced by the security objective O.REMOTE_SERVICE_AUDIT.

**OSP.ACTIVATION-KEY-ESCROW**
This OSP is enforced by the security objective OE.ACTIVATION-KEY-ESCROW.

**OSP.Atomic_Activation**
This OSP is enforced by the security objective O.Secure_AC_Activation and O.Secure_Load_ACode.

**OSP.TOE_Identification**
This OSP is enforced by the security objective O.TOE_Identification.

**OSP.Additional_Code_Signing**
This OSP is enforced by the security objectives O.Secure_Load_ACode and OE.Secure_ACode_Management.

### *7.3.4. Assumptions coverage – Rationale*

**A.MOBILE-OPERATOR**
This assumption is directly upheld by OE.MOBILE-OPERATOR.

**A.OTA-ADMIN**
This assumption is directly upheld by OE.OTA-ADMIN.

**A.APPS-PROVIDER**
This assumption is directly upheld by OE.APPS-PROVIDER.

**A.VERIFICATION-AUTHORITY**
This assumption is directly upheld by OE.VERIFICATIONAUTHORITY.

**A.CONTROLLING-AUTHORITY**
This assumption is directly upheld by OE.CONTROLLINGAUTHORITY.

**A.APPLET**
This assumption is upheld by the security objective for the operational environment OE.APPLET which ensures that no applet loaded post-issuance shall contain native methods.

**A.VERIFICATION**
This assumption is upheld by the security objective on the operational environment OE.CODE-EVIDENCE AND OE.VERIFICATION which guarantees that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time.

## 7.4. COMPOSITION TASKS – OBJECTIVES PART

### *7.4.1. Statement of compatibility – TOE Objectives part*

The following table (see next page) lists the relevant TOE security objectives of the ST33G1M2 IC, and provides the link to the composite-product TOE security objectives, showing that there is no contradiction between the two sets of objectives.

Note: The additional security objectives for the TOE defined in [ST_ST33G1M2] and related to DESFire are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security.

| Label of the chip TOE security objective | Title of the chip TOE security objective | Content of the chip TOE security objective | Linked Composite-product TOE security objectives |
|---|---|---|---|
| O.Leak-Inherent | Protection against Inherent Information Leakage | The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security IC<br>- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and<br>- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).<br>This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface. | O.SCP-SUPPORT<br>O.SCP.IC |
| O.Phys-Probing | Protection against Physical Probing | The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Security IC Embedded Software or against the disclosure of other critical information about the operation of the TOE. This includes protection against<br>- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or<br>- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis)<br>with a prior reverse-engineering to understand the design and its properties and functions. | O.SCP-SUPPORT<br>O.SCP.IC |
| O.Malfunction | Protection against Malfunctions | The TOE must ensure its correct operation.<br>The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields. | O.OPERATE |
| O.Phys-Manipulation | Protection against Physical Manipulation | The TOE must provide protection against manipulation of the TOE (including its software and Data), the Security IC Embedded Software and the User Data. This includes protection against<br>- reverse-engineering (understanding the design and its properties and functions),<br>- manipulation of the hardware and any data, as well as<br>- controlled manipulation of memory contents (Application Data). | O.SCP-SUPPORT<br>O.SCP.IC |
| O.Leak-Forced | Protection against Forced Information Leakage | The Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker<br>- by forcing a malfunction (refer to “Protection against Malfunction due to Environmental Stress (O.Malfunction)” and/or<br>- by a physical manipulation (refer to “Protection against Physical Manipulation (O.Phys-Manipulation)”. | O.SCP-SUPPORT<br>O.SCP.IC |

Copyright Gemalto SA – 2016

| Label of the chip TOE security objective | Title of the chip TOE security objective | Content of the chip TOE security objective | Linked Composite-product TOE security objectives |
|---|---|---|---|
| | | If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack. | |
| O.Abuse-Func | Protection against Abuse of Functionality | The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose critical User Data, (ii) manipulate critical User Data of the Security IC Embedded Software, (iii) manipulate Soft-coded Security IC Embedded Software or (iv) bypass, deactivate, change or explore security features or security services of the TOE. Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here. | O.SCP-SUPPORT |
| O.Identification | TOE Identification | The TOE must provide means to store Initialization Data and Pre-personalization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification. | No direct link to the composite-product TOE objectives, however chip traceability information stored in NVM is used by the TOE to answer identification CC requirements. |
| O.RND | Random Numbers | The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys. | O.RND |
| O.Add-Functions | Additional Specific Security Functionality | The TOE shall provide the following specific security functionality to the Security IC Embedded Software: Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES). | 3DES, AES: O.CIPHER<br>DES not used |
| O.Mem_Access | Dynamic Area based Memory Access Control | The TOE must provide the Security IC Embedded Software with the capability to define dynamic memory segmentation and protection. The TOE must then enforce the defined access rules so that access of software to memory areas is controlled as required, for example, in a multi-application environment. | O.SCP-SUPPORT |
| O.Controlled-ES-Loading | Controlled loading of the Security IC Embedded Software | The TOE must provide the capability to load the Security IC Embedded Software into the NVM, either before TOE delivery, under ST authority, either after TOE delivery, under the composite product manufacturer authority. The TOE must restrict the access to these features. The TOE must provide control means to check the integrity of the loaded user data. | This IC security objective supports the loading of the UpTeq NFC3.2.2_Generic v1.0 software during phase 6 (under Gemalto authority). |

### *7.4.2. Statement of compatibility – ENV Objectives part*

The following table lists the relevant ENV security objectives related to the ST33G1M2 chip, and provides the link to the composite-product, showing that they have been taken into account and that no contradiction has been introduced.

Note: The additional ENV security objectives defined in [ST_ST33G1M2] and related to DESFire are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security.

| IC ENV security objective label | IC ENV security objective title | IC ENV security objective content | Link to the composite-product |
|---|---|---|---|
| OE.Plat-Appl | Usage of Hardware Platform | To ensure that the TOE is used in a secure manner the Security IC Embedded Software shall be designed so that the requirements from the following documents are met:<br>– (i) hardware data sheet for the TOE,<br>– (ii) data sheet of the IC Dedicated Software of the TOE,<br>– (iii) TOE application notes, other guidance documents, and<br>– (iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report. | Fulfilled by the composite-SAR ADV_COMP.1<br>(cf [CCDB], Appendix 1.2, §72 and §73) |
| OE.Resp-Appl | Treatment of User Data | Security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context.<br>For example the Security IC Embedded Software will not disclose security relevant User Data to unauthorized users or processes when communicating with a terminal. | Covered by TOE Security Objectives:<br>O.COMM_AUTH, O.COMM_CONFIDENTIALITY<br>O.KEY-MNGT, O.PIN-MNGT |
| OE.Process-Sec-IC | Protection during composite product manufacturing | Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).<br>This means that Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately. | • During phases 4, 5 and 6: covered by the ALC composite-SARs and by the enforcement of OSP.PRODUCTION, OSP.PERSONALIZER and OSP.KEY-ESCROW.<br>• During phase 7, covered by all ENV objectives of the composite-TOE. |

Copyright Gemalto SA – 2016

# 8. Extended components definition

## 8.1. EXTENDED COMPONENT FCS_RND.1

### *8.1.1. Description*

The generation of random numbers requires that random numbers meet a defined quality metric.

### *8.1.2. Definition*

| **FCS_RND.1 Random Number Generation** |
|---|

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

Dependencies: No dependencies.

### *8.1.3. Rationale*

The FCS_RND family has been introduced initially by IC manufacturers to offer unpredictable random number generation; it is extended here to software platform. It defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

# 9. Security requirements

## 9.1. SECURITY FUNCTIONAL REQUIREMENTS

### *9.1.1. USIM Protection Profile*

**Card Manager (CMGRG)**

The security requirements below help to define a policy for controlling access to card content management operations and for expressing card issuer security concerns. Most of them come from [PP-JCS] but are instantiated to add more precisions regarding (U)SIM card content management.

Application note: patch management is an extension of the card management defined in GP since a patch is managed as a JavaCard Package, loaded as a standard executable load file and registered with specific attributes handled with GemActivate.

| **Card Content Management - Security Functional Requirements** |
|---|

| **FDP_UIT.1/CCM** | **Data exchange integrity** |
|---|---|

**FDP_UIT.1.1/CCM** The TSF shall enforce the **Secure Channel Protocol information flow control policy and the Security Domain access control policy** to **transmit and receive** user data in a manner protected from **modification, deletion, insertion and replay** errors.

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

Application note: patch is exchanged as a standard executable load file with specific attributes handled with GemActivate.

| **FDP_ROL.1/CCM** | **Basic rollback** |
|---|---|

**FDP_ROL.1.1/CCM** The TSF shall enforce **Security Domain access control policy** to permit the rollback of the **installation operation** on the **executable files and application instances (see application note)**.

**FDP_ROL.1.2/CCM** The TSF shall permit operations to be rolled back within the **size of the available memory when the card content management operation starts**.

Application note: patch is imported using such rules and loaded as a standard executable load file with specific attributes handled with GemActivate.

| **FDP_ITC.2/CCM** | **Import of user data with security attributes** |
|---|---|

**FDP_ITC.2.1/CCM** The TSF shall enforce the **Security Domain access control policy and the Secure Channel Protocol information flow policy** when importing user data, controlled under the SFP, from outside of the TOE.

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

**FDP_ITC.2.3/CCM** 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/CCM** 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/CCM** The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **The loading of a new Executable Load File is allowed only if, AID attribute of each dependent Executable File is equal to the identified AID in the CAP File, such AID is unique, SD is personalized and authorized to load. Otherwise, the load of ELF is rejected**.

Application notes:

- This Functional Component Instance enforces a security information flow control policy. Rules must be defined for importation operations. These rules must take into account all user data.
- Patch is also imported using such rules and loaded as a standard executable load file with specific attributes handled with GemActivate.

### FPT_FLS.1/CCM Failure with preservation of secure state

**FPT_FLS.1.1/CCM** The TSF shall preserve a secure state when the following types of failures occur: **the Security Domain fails to load/install an Executable File / application instance as described in GP22 §9.3.5**.

Application note: Patch is loaded as a standard executable load file with specific attributes handled with GemActivate.

### FCS_COP.1/DAP Cryptographic operation

**FCS_COP.1.1/DAP** The TSF shall perform **verification of the DAP signature attached to Executable Load Applications** in accordance with a specified cryptographic algorithm

- **PKC Scheme: SHA-1 hash and PKCS#1 RSA signature**
- **or DES Scheme: Single DES plus final Triple DES MAC (Retail MAC)**

and cryptographic key sizes

- **PKC Scheme: RSA key of minimum length 1024 bits**
- **DES Scheme: DES key of minimum length 16 byte**s

that meet the following:

- **Sections C.1.2 and C.6 of [GP]**
- **PKC Scheme: SSA-PKCS1-v1_5 as defined in PKCS#1**
- **DES Scheme: ISO 9797-1 as MAC Algorithm 3 with output transformation 3, without truncation, and with DES taking the place of the block cipher**

## Security Domain - Security Functional Requirements

Application note: Rules for security domain are also applicable for patch. A patch is managed as a JavaCard package and loaded as a standard executable load file with specific attributes handled with GemActivate.

### FDP_ACC.1/SD Subset access control

**FDP_ACC.1.1/SD** The TSF shall enforce the **Security Domain access control policy** on:

- **Subjects: S.INSTALLER, S.ADEL, S.CAD (from [PP-JCS]) and S.SD**
- **Objects: Delegation Token, DAP Block and Load File**
- **Operations: GlobalPlatform's card content management APDU commands and API methods**.

| FDP_ACF.1/SD | Security attribute based access control |
|---|---|

**FDP_ACF.1.1/SD** The TSF shall enforce the **Security Domain access control policy** to objects based on the following:

- **Subjects:**
  - **S.INSTALLER, defined in [PP-JCS] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [GP]);**
  - **S.ADEL, also defined in [PP-JCS] and represented by the GlobalPlatform Environment (OPEN) on the card;**
  - **S.SD receiving the Card Content Management commands (through APDUs or APIs) with a set of privileges (defined in Section 6.6.1 of [GP]), a life-cycle status (defined in Section 5.3.2 of [GP]) and a Secure Communication Security level (defined in Section 10.6 of [GP]);**
  - **S.CAD, defined in [PP-JCS], the off-card entity that communicates with the S.INSTALLER through S.SD;**
- **Objects:**
  - **The Delegation Token, in case of Delegated Management operations, with the attributes Present or Not Present;**
  - **The DAP Block, in case of application loading, with the attributes Present or Not Present;**
  - **The Load File or Executable File, in case of application loading, installation, extradition or registry update, with a set of intended privileges and its targeted associated SD AID.**
- **The following security attributes:**
  - **The Default Selected attribute specifies whether the applet instance is the one that should be executed when no application has been explicitly selected.**
  - **The Application State attribute specifies the current life cycle state of the application instance, which may be either SELECTABLE, APPLICATION_SPECIFIC, LOCKED.**
  - **The Card State attribute, is the current state in the life cycle of the card, which may be either OP_READY, INITIALIZED, SECURED, CARD_LOCKED, or TERMINATED.**
  - **The Card Lock attribute specifies whether the applet is allowed to temporary lock the services of the smart card.**
  - **The Card Termination attribute specifies whether the applet is allowed to definitely disable the services of the smart card.**
  - **The CVM attribute specifies whether the applet is allowed to modify the try limit and the PIN code of the global CVM service.**
  - **The Registered Applications attribute specifies the Executable Files and application instances that have been installed on the card so far and their dependencies**.

**FDP_ACF.1.2/SD** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **Runtime behavior rules defined by GlobalPlatform for:**

- **loading (Section 9.3.5 of [GP]);**
- **installation (Section 9.3.6 of [GP]);**
- **extradition (Section 9.4.1 of [GP]);**
- **registry update (Section 9.4.2 of [GP]);**
- **content removal (Section 9.5 of [GP])**.

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

- **Rule SD-1: A card administration request may be accepted only if the APDU command specifying the request is well-formed according to [GP22].**
- **Rule SD-2: A card administration request other than requesting card management data may be accepted only if the Card State is not TERMINATED.**
- **Rule SD-3: The selection of an applet instance may be accepted only if the Applet State is not LOCKED.**

- **Rule SD-4: The update of the life cycle state of an application instance is accepted only if the new state is consistent with its current life cycle state according to GlobalPlatform's life cycle rules (either coming from an APDU command or from an application instance through the GP API).**
- **Rule SD-5: A request for installing an Executable Load File may be accepted only if there is enough resources for loading the Executable File, and no Executable File on the card has been already registered with the specified AID.**
- **Rule SD-6: A Executable Load File block may be loaded only if all its previous blocks have been received in order, and there are sufficient resources for storing the new one.**
- **Rule SD-7: A new applet instance may be created only if the Package Properties enables applet instantiation or multiple applet instances (if there is already an instance for that applet) but also if the AID specified for the applet instance is not already used for another applet or Executable File installed on the card, and the privileges specified for it are consistent with the GlobalPlatform rules specified in [VGP].**
- **Rule SD-8: An Executable File may be deleted from the smart card only if it is not reachable from other Executable Files or application instances on the card.**
- **Rule SD-9: An applet instance may be deleted from the card only it is not currently active on a logical channel, and none of the resources it has allocated is reachable from other Executable Files or Application instances installed on the card.**
- **Rule SD-10: An applet instance may lock the card only if it has the Card Lock privilege.**
- **Rule SD-11: An applet instance may terminate the card only if it has the Card Termination privilege.**
- **Rule SD-12: An applet instance may unlock the CVM service or modify the CVM try limit or PIN code only if it has the CVM privilege.**
- **Rule SD-13: A request involving the use of any of the Security domain keys is accepted only if the concerned keys are integer**.

**FDP_ACF.1.4/SD** The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **when at least one of the rules defined by GlobalPlatform does not hold**.

**FMT_MSA.1/SD Management of security attributes**

**FMT_MSA.1.1/SD** The TSF shall enforce the **Security Domain access control policy** to restrict the ability to **modify** the security attributes **Any security attributes registered the GP Registry such as:**

- **Application state of an application instance (1)**
- **Default selected application (2)**
- **Card Life cycle state (3)**
- **Package properties (4)**
- **Application association (5)**

to

- **the Security Domain and the application instance itself (1)**
- **the Security Domain (2&4)**
- **the Security Domain and application with privilege (Card Lock or Terminated) (3)**.

**FMT_MSA.3/SD Static attribute initialization**

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

**FMT_MSA.3.2/SD** The TSF shall allow the **Issuer or authorized application provider** to specify alternative initial values to override the default values when an object or information is created.

Refinement:

- Alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1.
- The Default Selected application shall be the ISD.
- The initial value of the Application State of an applet instance shall be SELECTABLE.

Application notes:

When the TOE enters the life cycle phases under the scope of this Security Target, the Card State shall be at least SECURED. The initial value of the Application State of an applet instance shall be SELECTABLE. The initial Package Properties shall enable all card content management operations on the package.

The Issuer or authorized application provider may assign the Default Select privilege to another application instance.

| FMT_SMF.1/SD | Specification of Management Functions |
|---|---|

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

- **Restricting the properties associated to a given package**
- **Registering a new Executable File or application instance in the GP registry.**
- **Removing the specified entries from the GP registry when a DELETE command is received.**
- **Unsetting it as the Default Select application and set this privilege to a new application instance.**
- **Granting the privileges that the authorized entities (MNO, or Application Provider) specifies when a new application instance is installed**.

| FMT_SMR.1/SD | Security roles |
|---|---|

**FMT_SMR.1.1/SD** The TSF shall maintain the roles

- **Issuer Security Domain**
- **Supplementary Security Domain**
- **Certification Authority Security Domain**.
- **Verification Authority Security Domain**

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

**Secure Channel - Security Functional Requirements**

Application note: Rules for security channel are also applicable for patch. A patch is managed as a JavaCard package and loaded as a standard executable load file.

| FTP_ITC.1/SC | Inter-TSF trusted channel |
|---|---|

**FTP_ITC.1.1/SC** The TSF shall provide a communication channel between itself and another 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/SC** The TSF shall permit **another trusted IT product** to initiate communication via the trusted channel.

**FTP_ITC.1.3/SC** The TSF shall initiate communication via the trusted channel for **all card management functions:**

- **loading or deleting an Executable Load file;**
- **installing or removing an application instance;**
- **extrading an Executable Load file or an application instance;**
- **registry update;**
- **Loading or removing a Key Set;**
- **SD personalization;**
- **Changing the Application Life Cycle or card Life Cycle.**

| FCO_NRO.2/SC | Enforced proof of origin |
|---|---|

**FCO_NRO.2.1/SC** The TSF shall enforce the generation of evidence of origin for transmitted **Executable load files** at all times.

**FCO_NRO.2.2/SC** The TSF shall be able to relate the **identity** of the originator of the information, and the **Executable Load Files** of the information to which the evidence applies.

**FCO_NRO.2.3/SC** The TSF shall provide a capability to verify the evidence of origin of information to **originator** given **Executable load files**.

| FDP_IFC.2/SC | Complete information flow control |
|---|---|

**FDP_IFC.2.1/SC** The TSF shall enforce the **Secure Channel Protocol information flow control policy** on

- **the subjects S.CAD and S.SD, involved in the exchange of messages between the (U)SIM card and the CAD through a potentially unsafe communication channel**
- **the information controlled by this policy is the card content management command, including personalization commands, in the APDUs sent to the card and their associated responses returned to the CAD.**

**The subjects covered by this policy are those involved in the exchange of messages between the card and the CAD through a potentially unsafe communication channel:**

- **An off-card subject that represents the authorized entities (S.BCV).**
- **Any application with the Security Domain privilege (S.CRD).**

**The information controlled by this policy is the one contained in the APDU commands sent to the card and their associated responses returned to the CAD or the mobile.**

and all operations that cause that information to flow to and from subjects covered by the SFP.

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

| FDP_IFF.1/SC | Simple security attributes |
|---|---|

**FDP_IFF.1.1/SC** The TSF shall enforce the **Secure Channel Protocol information flow control policy** based on the following types of subject and information security attributes:

- **Subjects:**
  - **S.SD receiving the Card Content Management commands (through APDUs or APIs). This subject can be the ISD, an APSD or a CASD.**
  - **S.CAD the off-card entity that communicates with the S.SD.**
- **Information:**
  - **load file, in case of application loading;**
  - **applications or SD privileges, in case of application installation or registry update;**
  - **personalization keys and/or certificates, in case of application or SD personalization.**

**The subjects have the following security attributes for SCP02 [GP]:**

- **The Challenge is a random number generated by the subject in order to identify the current session.**
- **The Cryptogram is a secret relative to the current smart card session that serves to authenticate the on- and off-card subjects. The cryptogram is derived from the challenges of both the card and the terminal.**
- **The Key Set is a collection of three keys (Secure Channel Encryption Key (S-ENC), a Command Message Authentication Code Key (C-MAC) and a Data Encryption Key (DEK)) used to encrypt the Derivation Data in order to generate the session keys. It is identified by a key version number.**
- **The Session Keys is a set of keys derived from KeySet and sequence counter to be used to verify the origin and integrity of the received message, and to decrypt their contents. This set is made of the following keys:**
  - **Command Message Authentication Code Key (C-MAC session key);**

  - Encryption Key (S-ENC session key);
  - Data Encryption Key (DEK session key).
- The Command Security Level defined for the messages that the card receives through the secure channel. The possible security levels are:
  - NO-SEC (clear text),
  - C-AUTHENTICATED (authentication of the command's issuer),
  - C-MAC (authentication of the issuer and integrity of the command),
  - C-DEC (authentication of the issuer, integrity and confidentiality of the command).
- The Initial Chaining Vector (ICV) is a value used to compute the MAC value of a message, which relates it to the previous messages of the current session.

The subjects have the following security attributes for SCP03 [GP-SCP]:

- The Challenge is a random number generated by the subject in order to identify the current session.
- The Cryptogram is a secret relative to the current smart card session that serves to authenticate the on- and off-card subjects. The cryptogram is derived from the challenges of both the card and the terminal.
- The Key Set is a collection of three keys (Static Secure Channel Encryption Key (Key-ENC), Static Secure Channel Message Authentication Code Key (Key-MAC) and a Data Encryption Key (Key-DEK)) used to encrypt the Derivation Data in order to generate the session keys. It is identified by a key version number.
- The Session Keys is a set of keys derived from KeySet and sequence counter to be used to verify the origin and integrity of the received message, and to decrypt their contents. This set is made of the following keys:
  - Secure Channel Message Authentication Code Key for Command (S-MAC);
  - Secure Channel Message Authentication Code Key for Response (S-RMAC);
  - Session Secure Channel Encryption Key (S-ENC).
- The Command Security Level defined for the messages that the card receives through the secure channel. The possible security levels are:
  - NO-SEC (clear text),
  - AUTHENTICATED (authentication of the command's issuer),
  - C-MAC (authentication of the issuer and integrity of the command),
  - C-DECRYPTION (authentication of the issuer, integrity and confidentiality of the command),
  - R-MAC (authentication of the card and integrity of the response),
  - R-ENCRYPTION (authentication of the card, integrity and confidentiality of the response).
- The Initial Chaining Vector (ICV) is a value used to compute the MAC value of a message, which relates it to the previous messages of the current session.

The security attributes for SCP80 are:

- CPL, CHL giving Information about the length of the received message and the length of the security header in that message;
- SPI containing the security level applied to the incoming message, and response (if any), defining properties for message integrity and authentication, replay detection and sequence integrity and confidentiality;
- TAR (Target Application Reference), indicating the application which the message is addressed to;
- KIC and KID both contain information on the keys (key set number and the key algorithm) to be used when checking the security;
- CNTR, a synchronization counter to avoid playing the same message several times;
- PCNTR, padding information used only when the message is encrypted;
- and finally a signature, that can be either a basic CRC of the message or a signature involving keys.

The security attributes for SCP81 as defined in [GP-RAM] §4.7 are: security domain parameters, connection parameters, security parameters, retry policy parameters, HTTP POST parameters.

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

- **Runtime behavior rules defined by GlobalPlatform for:**
  - **loading (Section 9.3.5 of [GP]);**
  - **installation (Section 9.3.6 of [GP]);**
  - **extradition (Section 9.4.1 of [GP]);**
  - **registry update (Section 9.4.2 of [GP]);**
  - **SD personalization rules, pull and push models (Section 11 of [GP-UICC]).**
- **Rule IFF-1: The SD may process a RECEIVE (INITIALIZE-UPDATE) operation only if the key set specified in the command exist in the SD and is integer.**
  **Rule IFF-2: The ISD may process a RECEIVE (EXTERNAL-AUTHENTICATE) operation if the following conditions hold:**
  - **The cryptogram received from the off-card subject is equal to the cryptogram computed by the Security Domain.**
  - **The MAC attached to the message has been generated from the CMAC session key and the current value of the ICV.**

  **Rules IFF-3: The ISD may process a RECEIVE (GET-DATA) operation if the following condition holds: If the command security level is at least C-MAC, the MAC attached to the message has been generated from the command using the C-MAC session key and the current value of the ICV.**
  **Rules IFF-4: The ISD may process a RECEIVE (M) operation for any other command M different from the ones cited in the rules above if the following conditions hold:**
  - **The current security level is at least AUTHENTICATED.**
  - **If the command security level is at least C-MAC, the MAC attached to the message has been generated from the clear-text command using the C-MAC session key and the current value of the ICV**.

**FDP_IFF.1.3/SC** The TSF shall enforce the **no additional information flow control SFP rules**.

**FDP_IFF.1.4/SC** The TSF shall explicitly authorize an information flow based on the following rules: **no additional information flow control SFP rules**.

**FDP_IFF.1.5/SC** The TSF shall explicitly deny an information flow based on the following rules: **When none of the conditions listed in the element FDP_IFF.1.4 of this component hold and at least one of those listed in the element FDP_IFF.1.2 does not hold**.

**FMT_MSA.1/SC Management of security attributes**

**FMT_MSA.1.1/SC** The TSF shall enforce the **Secure Channel Protocol (SCP) information flow control policy** to restrict the ability to **modify** the security attributes **(1) key set, Static keys, Command security Level, Secure channel protocol of a security domain, (2) Session Keys, Sequence Counter and ICV of a session (for SCP02 and SCP03), (3) SPI, TAR, CNTR, PCNTR, signature (for SCP80), (4) security domain parameters, connection parameters, security parameters, retry policy parameters, HTTP POST parameters (for SCP81),** to **(1 & 2 & 3 & 4) the actor associated with the security domain:**

- **The MNO for ISD,**
- **The Service Provider for SSD**,
- **The CA for CASD.**

Application note: the authorized identified roles could be the card issuer (off-card) or a SD (on-card).

**FMT_MSA.3/SC Static attribute initialization**

**FMT_MSA.3.1/SC** The TSF shall enforce the **Secure Channel Protocol (SCP) information flow control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

**FMT_MSA.3.2/SC** The TSF shall allow the **authorized entities (MNO, or Application Provider)** to specify alternative initial values to override the default values when an object or information is created.

Refinement: alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1.

**FMT_SMF.1/SC Specification of Management Functions**

**FMT_SMF.1.1/SC** The TSF shall be capable of performing the following management functions:
**Management functions specified in GlobalPlatform specifications [GP]:**

- **loading (Section 9.3.5 of [GP]);**
- **installation (Section 9.3.6 of [GP]);**
- **extradition (Section 9.4.1 of [GP]);**
- **registry update (Section 9.4.2 of [GP]);**
- **SD personalization rules, pull and push models (Section 11 of [GP-UICC]).**

**The management functions are:**

- **For SCP02 and SCP03**
  - **Generating a new card challenge during the setup of a Secure Channel.**
  - **Generating the session keys for the Secure Channel from the specified static key set and its associated Sequence Counter.**
  - **Generating the card cryptogram from the host and card challenges and the session keys.**
  - **Increasing by one the Sequence Counter associated to the specified Key Set upon successful opening a Secure Channel.**
  - **Setting the security level of the Secure Channel as the authenticated authorized entities (MNO, or Application Provider) had specified during its setup.**
  - **Updating the current value of the ICV upon reception of a new message through the Secure Channel.**
  - **On request of the Issuer or authorized application provider, loading or replacing the static keys that the associated Security Domain uses to open a Secure Channel.**
- **For SCP80:**
  - **modifying parameter values (CPL, CHL, SPI, Kic, Kid, TAR, CNTR, PCNTR, signature),**
  - **Setting the security level of the Secure Channel as the authenticated authorized entities (MNO, or Application Provider) had specified during its setup,**
  - **On request of the Issuer or authorized application provider, loading or replacing the static keys that the associated Security Domain uses to open a Secure Channel**.
- **For SCP81:**
  - **initiating a TLS session with handshake parameters defined in "ClientHello" message (random number A and supported cipher suites*) and received in remote server answers in "ServerHello" and "ServerHelloDone" messages (random number B and session identifier) and answering with "ClientKeyExchange" and "ChangeCipherSpec" messages with the PSK identity to use and the chosen Cipher Suite, receiving remote server answers with "ChangeCipherSpec" and "Finished" messages,**
  - **setting parameter values for: security domain parameters, connection parameters, security parameters, retry policy parameters, HTTP POST parameters**

Application note:

- All management functions related to SCP02 secure channel shall be relevant.
- Supported ciphered suites in SCP81 are: TLS_PSK_WITH_3DES_EDE_CBC_SHA, TLS_PSK_WITH_AES_128_CBC_SHA, as defined in [RFC4279] and TLS_PSK_WITH_NULL_SHA as defined in [RFC4785].

| FIA_UID.1/SC | Timing of identification |
|---|---|

**FIA_UID.1.1/SC** The TSF shall allow

- **application selection;**
- **initializing a secure channel with the card;**
- **requesting data that identifies the card or the Card Issuer;**

on behalf of the user to be performed before the user is identified.

**FIA_UID.1.2/SC** 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 GlobalPlatform TSF mediated actions listed in [GP] such as selecting an application, requesting data, initializing, etc.

| FIA_UAU.1/SC | Timing of authentication |
|---|---|

**FIA_UAU.1.1/SC** The TSF shall allow **the TSF mediated actions listed in FIA_UID.1/SC** on behalf of the user to be performed before the user is authenticated.

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

| FIA_UAU.4/SC | Single-use authentication mechanisms |
|---|---|

**FIA_UAU.4.1/SC** The TSF shall prevent reuse of authentication data related to **the authentication mechanism used to open a secure communication channel with the card**.

### *9.1.2. JCS Protection Profile*

This section states the security functional requirements for the Java Card System - Open configuration. For readability, requirements are arranged into groups. All the groups defined in the table below come from [PP-JCS].

| Group | Name | Description |
|---|---|---|
| CoreG_LC | Core with Logical Channels | The CoreG_LC contains the requirements concerning the runtime environment of the Java Card System implementing logical channels. This includes the firewall policy and the requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2 feature. This group is the union of requirements from the Core (CoreG) and the Logical channels (LCG) groups defined in [PP/0305] (cf. Java Card System Protection Profile Collection [PP JCS]). |
| InstG | Installation | The InstG contains the security requirements concerning the installation of post-issuance applications. It does not address card management issues in the broad sense, but only those security aspects of the installation procedure that are related to applet execution. |
| ADELG | Applet deletion | The ADELG contains the security requirements for erasing installed applets from the card, a feature introduced in Java Card specification version 2.2. |
| ODELG | Object deletion | The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism. This is a Java Card specification version 2.2 feature. |
| CarG | Secure Carrier | The CarG group contains minimal requirements for secure downloading of applications on the card. This group contains the security requirements for preventing, in those configurations that do not support on-card static or dynamic bytecode verification, the installation of a package that has not been bytecode verified, or that has been modified after bytecode verification. |

Subjects are active components of the TOE that (essentially) act on the behalf of users. The users of the TOE include people or institutions (like the applet developer, the card issuer, the verification authority), hardware (like the CAD where the card is inserted or the PCD) and software components (like the application packages installed on the card). Some of the users may just be aliases for other users. For instance, the verification authority in charge of the bytecode verification of the applications may be just an alias for the card issuer.

Subjects (prefixed with an "S") are described in the following table:

| Subject | Description |
|---|---|
| S.ADEL | The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([JCRE301], §11), but its role asks anyway for a specific treatment from the security viewpoint. This subject is unique and is involved in the ADEL security policy defined in §7.1.3.1. |
| S.APPLET | Any applet instance. |
| S.BCV | The bytecode verifier (BCV), which acts on behalf of the verification authority who is in charge of the bytecode verification of the packages. This subject is involved in the PACKAGE LOADING security policy defined in §7.1.7. |
| S.CAD | The CAD represents the actor that requests, by issuing commands to the card. It also plays the role of the off-card entity that communicates with the S.INSTALLER. |
| S.INSTALLER | The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of packages and installation of applets. |
| S.JCRE | The runtime environment under which Java programs in a smart card are executed. |
| S.JCVM | The bytecode interpreter that enforces the firewall at runtime. |
| 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. |
| S.PACKAGE | A package is a namespace within the Java programming language that may contain classes and interfaces, and in the context of Java Card technology, it defines either a user library, or one or several applets. |

Objects (prefixed with an "O") are described in the following table:

| Object | Description |
|---|---|
| O.APPLET | Any installed applet, its code and data. |
| O.CODE_PKG | The code of a package, including all linking information. On the Java Card platform, a package is the installation unit. |
| O.JAVAOBJECT | Java class instance or array. It should be noticed that KEYS, PIN, arrays and applet instances are specific objects in the Java programming language. |

Information (prefixed with an "I") is described in the following table:

| Information | Description |
|---|---|
| I.APDU | Any APDU sent to or from the card through the communication channel. |
| I.DATA | JCVM Reference Data: objectref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method. |

Security attributes linked to these subjects, objects and information are described in the following table with their values:

| Security attribute | Description / Value |
|---|---|
| Active Applets | The set of the active applets' AIDs. An active applet is an applet that is selected on at least one of the logical channels. |
| Applet Selection Status | "Selected" or "Deselected". |
| Applet's version number | The version number of an applet (package) indicated in the export file. |
| Context | Package AID or "Java Card RE". |
| Currently Active Context | Package AID or "Java Card RE". |
| Dependent package AID | Allows the retrieval of the Package AID and Applet's version number ([JCVM301], §4.5.2). |

| | |
|---|---|
| LC Selection Status | Multiselectable, Non-multiselectable or "None". |
| LifeTime | CLEAR_ON_DESELECT or PERSISTENT (*). |
| Owner | 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). |
| Package AID | The AID of each package indicated in the export file. |
| Registered Applets | The set of AID of the applet instances registered on the card. |
| Resident Packages | The set of AIDs of the packages already loaded on the card. |
| Selected Applet Context | Package AID or "None". |
| Sharing | Standards, SIO, Java Card RE entry point or global array. |
| Static References | Static fields of a package may contain references to objects. The Static References attribute records those references. |

(*) Transient objects of type CLEAR_ON_RESET behave like persistent objects in that they can be accessed only when the Currently Active Context is the object's context.

Operations (prefixed with "OP") are described in the following table. Each operation has 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.

| Operation | Description |
|---|---|
| OP.ARRAY_ACCESS (O.JAVAOBJECT, field) | Read/Write an array component. |
| OP.CREATE(Sharing, LifeTime) (*) | Creation of an object (new or makeTransient call). |
| OP.DELETE_APPLET(O.APPLET,...) | Delete an installed applet and its objects, either logically or physically. |
| OP.DELETE_PCKG(O.CODE_PKG,...) | Delete a package, either logically or physically. |
| OP.DELETE_PCKG_APPLET (O.CODE_PKG,...) | Delete a package and its installed applets, either logically or physically. |
| OP.INSTANCE_FIELD (O.JAVAOBJECT, field) | Read/Write a field of an instance of a class in the Java programming language. |
| OP.INVK_VIRTUAL (O.JAVAOBJECT, method, arg1,...) | Invoke a virtual method (either on a class instance or an array object). |
| OP.INVK_INTERFACE (O.JAVAOBJECT, method, arg1,...) | Invoke an interface method. |
| OP.JAVA(...) | Any access in the sense of [JCRE301], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS. |
| OP.PUT(S1,S2,I) | Transfer a piece of information I from S1 to S2. |
| OP.THROW(O.JAVAOBJECT) | Throwing of an object (athrow, see [JCRE301], §6.2.8.7). |
| OP.TYPE_ACCESS (O.JAVAOBJECT, class) | Invoke checkcast or instanceof on an object in order to access to classes (standard or shareable interfaces objects). |

(*) For this operation, there is no accessed object. This rule enforces that shareable transient objects are not allowed. For instance, during the creation of an object, the JavaCardClass attribute's value is chosen by the creator.

### CoreG_LC Security Functional Requirements

This group is focused on the main security policy of the Java Card System, known as the firewall.

**Firewall Policy**

#### 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, S.JCVM, O.JAVAOBJECT** and all operations among subjects and objects covered by the SFP.

Refinement: the operations involved in the policy are: OP.CREATE, OP.INVK_INTERFACE, OP.INVK_VIRTUAL, OP.JAVA, OP.THROW, OP.TYPE_ACCESS.

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

Application note: It should be noticed 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_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:

| Subject / Object | Security attributes |
|---|---|
| S.PACKAGE | LC Selection Status |
| S.JCVM | Active Applets, Currently Active Context |
| S.JCRE | Selected Applet Context |
| O.JAVAOBJECT | Sharing, Context, LifeTime |

**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:

- **R.JAVA.1 ([JCRE22], §6.2.8): S.PACKAGE may freely perform 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): S.PACKAGE may freely perform 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): 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.4 ([JCRE22], §6.2.8.6): 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 the invoked interface method extends the Shareable interface and one of the following conditions 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.**
- **R.JAVA.5: S.PACKAGE may perform OP.CREATE only if the value of the Sharing parameter is "Standard"**.

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

1) **The subject S.JCRE can freely perform OP.JAVA(") and OP.CREATE, with the exception given in FDP_ACF.1.4/FIREWALL, provided it is the Currently Active Context.**
2) **The only means that the subject S.JCVM shall provide for an application to execute native code is the invocation of a Java Card API method (through OP.INVK_INTERFACE or OP.INVK_VIRTUAL).**

**FDP_ACF.1.4/FIREWALL** The TSF shall explicitly deny access of subjects to objects based on the following additional 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 attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the Selected Applet Context.**

Application note, FDP_ACF.1.4/FIREWALL:

The deletion of applets may render some O.JAVAOBJECT inaccessible, and the Java Card RE may be in charge of this aspect. This can be done, for instance, by ensuring that references to objects belonging to a deleted application are considered as a null reference. Such a mechanism is implementation-dependent.

In the case of an array type, fields are components of the array ([JVM], §2.14, §2.7.7), as well as the length; the only methods of an array object are those inherited from the Object class.

The Sharing attribute defines four categories of objects:

- Standard ones, whose both fields and methods are under the firewall policy,
- Shareable interface Objects (SIO), which provide a secure mechanism for inter-applet communication,
- JCRE entry points (Temporary or Permanent), who have freely accessible methods but protected fields,
- Global arrays, having both unprotected fields (including components; refer to JavaCardClass discussion above) and methods.

When a new object is created, it is associated with the Currently Active Context. But the object is owned by the applet instance within the Currently Active Context when the object is instantiated ([JCRE22], §6.1.3). An object is owned by an applet instance, by the JCRE or by the package library where it has been defined (these latter objects can only be arrays that initialize static fields of packages).

([JCRE22], Glossary) Selected Applet Context. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command with this applet's AID, the Java Card RE makes this applet the Selected Applet Context. The Java Card RE sends all APDU commands to the Selected Applet Context.

While the expression "Selected Applet Context" refers to a specific installed applet, the relevant aspect to the policy is the context (package AID) of the selected applet. In this policy, the "Selected Applet Context" is the AID of the selected package.

([JCRE22], §6.1.2.1) At any point in time, there is only one active context within the Java Card VM (this is called the Currently Active Context).

It should be noticed that the invocation of static methods (or access to a static field) is not considered by this policy, as there are no firewall rules. They have no effect on the active context as well and the "acting package" is not the one to which the static method belongs to in this case.

It should be noticed that the Java Card platform, version 2.2.x and version 3 Classic Edition, 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 ([JCVM22], §2.2.5). Therefore, the selection mode can be regarded as an attribute of packages. No selection mode is defined for a library package.

An applet instance will be considered an active applet instance if it is currently selected in at least one logical channel. 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).

| FDP_IFC.1/JCVM | Subset information flow control |
|---|---|

**FDP_IFC.1.1/JCVM** The TSF shall enforce the **JCVM information flow control SFP** on **S.JCVM, S.LOCAL, S.MEMBER, I.DATA and OP.PUT(S1, S2, I)**.

Application note: it should be noticed that references of temporary Java Card RE entry points, which cannot be stored in class variables, instance variables or array components, are transferred from the internal memory of the Java Card RE (TSF data) to some stack through specific APIs (Java Card RE owned exceptions) or Java Card RE invoked methods (such as the process (APDU apdu)); these are causes of OP.PUT(S1,S2,I) operations as well.

| 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:

| Subjects | Security attributes |
|---|---|
| S.JCVM | 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.DATA) is allowed if and only if the Currently Active Context is "Java Card RE";**
- **Other OP.PUT operations are allowed regardless of the Currently Active Context's value**.

**FDP_IFF.1.3/JCVM** The TSF shall enforce the **[No additional rules]**.

**FDP_IFF.1.4/JCVM** The TSF shall explicitly authorize an information flow based on the following rules: **[No additional rules]**.

**FDP_IFF.1.5/JCVM** The TSF shall explicitly deny an information flow based on the following rules: **[No additional rules]**.

Application note:

The storage of temporary Java Card RE-owned objects references is runtime-enforced ([JCRE22], §6.2.8.1-3).

It should be noticed that this policy essentially applies to the execution of bytecode. Native methods, the Java Card RE itself and possibly some API methods can be granted specific rights or limitations through the FDP_IFF.1.3/JCVM to FDP_IFF.1.5/JCVM elements. The way the Java Card virtual machine manages the transfer of values on the stack and local variables (returned values, uncaught exceptions) from and to internal registers is implementation-dependent. For instance, a returned reference, depending on the implementation of the stack frame, may transit through an internal register prior to being pushed on the stack of the invoker. The returned bytecode would cause more than one OP.PUT operation under this scheme.

**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 initialized with default values when the resource is allocated [JVM], §2.5.1.

**FMT_MSA.1/JCRE Management of security attributes**

**FMT_MSA.1.1/JCRE** The TSF shall enforce the **FIREWALL access control SFP** to restrict the ability to **modify** the security attributes **Selected Applet Context** to **the Java Card RE**.

Application note: the modification of the Selected Applet Context should be performed in accordance with the rules given in [JCRE22], §4 and [JCVM22], §3.4.

**FMT_MSA.1/JCVM Management of security attributes**

**FMT_MSA.1.1/JCVM** 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 **Currently Active Context and Active Applets** to **the Java Card VM (S.JCVM)**.

Application note: the modification of the Currently Active Context should be performed in accordance with the rules given in [JCRE22], §4 and [JCVM22], §3.4.

**FMT_MSA.2/FIREWALL_JCVM Secure security attributes**

**FMT_MSA.2.1/FIREWALL_JCVM** The TSF shall ensure that only secure values are accepted for **all the security attributes of subjects and objects defined in the FIREWALL access control SFP and the JCVM information flow control SFP**.

Application note: the following rules are given as examples only. For instance, the last two rules are motivated by the fact that the Java Card API defines only transient arrays factory methods. Future versions may allow the creation of transient objects belonging to arbitrary classes; such evolution will naturally change the range of "secure values" for this component.

- The Context attribute of an O.JAVAOBJECT must correspond to that of an installed applet or be "Java Card RE".
- An O.JAVAOBJECT whose Sharing attribute is a Java Card RE entry point or a global array necessarily has "Java Card RE" as the value for its Context security attribute.
- An O.JAVAOBJECT whose Sharing attribute value is a global array necessarily has "array of primitive type" as a 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 initialization**

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

**FMT_MSA.3.2/FIREWALL** **[Editorially Refined]** The TSF shall not allow **any role** to specify alternative initial values to override the default values when an object or information is created.

Application note, FMT_MSA.3.1/FIREWALL :

Objects' security attributes of the access control policy are created and initialized at the creation of the object or the subject. Afterwards, these attributes are no longer mutable (FMT_MSA.1/JCRE). At the creation of an object (OP.CREATE), the newly created object, assuming that the FIREWALL access control SFP permits the operation, gets its Lifetime and Sharing attributes from the parameters of the operation; on the contrary, its Context attribute has a default value, which is its creator's Context attribute and AID respectively ([JCRE22], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is "Java Card RE".

The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

Application note, FMT_MSA.3.2/FIREWALL :

The intent is that none of the identified roles has privileges with regard to the default values of the security attributes. It should be noticed that creation of objects is an operation controlled by the FIREWALL access control SFP. The operation shall fail anyway if the created object would have had security attributes whose value violates FMT_MSA.2.1/FIREWALL_JCVM.

**FMT_MSA.3/JCVM Static attribute initialization**

**FMT_MSA.3.1/JCVM** The TSF shall enforce 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/JCVM [Editorially Refined]** The TSF shall not allow **any role** to specify alternative initial values to override the default values when an object or information is created.

**FMT_SMF.1 Specification of Management Functions**

**FMT_SMF.1.1** The TSF shall be capable of performing the following management functions: **modify the Currently Active Context, the Selected Applet Context and the Active Applets**.

**FMT_SMR.1 Security roles**

**FMT_SMR.1.1** The TSF shall maintain the roles**:**

- **Java Card RE (JCRE),**
- **Java Card VM (JCVM)**.

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

---

**Application Programming Interface**

The following SFRs are related to the Java Card API.

The whole set of cryptographic algorithms is generally not implemented because of limited memory resources and/or limitations due to exportation. Therefore, the following requirements only apply to the implemented subset.

It should be noticed that the execution of the additional native code is not within the TSF. Nevertheless, access to API native methods from the Java Card System is controlled by TSF because there is no difference between native and interpreted methods in their interface or invocation mechanism.

**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)**.

Application note: the keys are generated and diversified in accordance with [JCAPI301] in class KeyBuilder (buildKey method).

**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, 192 and 256 bits** that meet the following: **none (random numbers generation)**.

Application note: the keys are generated and diversified in accordance with [JCAPI301] in class KeyBuilder (buildKey method).

**FCS_CKM.1/RSA Cryptographic key generation**

**FCS_CKM.1.1/RSA** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA Standard and RSA CRT Key Pair Generation** and specified cryptographic key sizes **512 to 2048 bits by steps of 32 bits** that meet the following: **see application note**.

Application note: the keys are generated and diversified in accordance with [JCAPI301] in classes KeyBuilder (buildKey method) and KeyPair (genKeyPair method).

**FCS_CKM.1/ECDSA Cryptographic key generation**

**FCS_CKM.1.1/ECDSA** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **ECDSA Key Pair Generation** and specified cryptographic key sizes **[P ranging from 192 to 521 bits]** that meet the following: **see application note**.

Application note: the keys are generated and diversified in accordance with [JCAPI301] in classes KeyBuilder (buildKey method) and KeyPair (genKeyPair method).

**FCS_CKM.1/HMAC Cryptographic key generation**

**FCS_CKM.1.1/HMAC** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **HMAC Key generation** and specified cryptographic key sizes **[see application note]** that meet the following: **none (random numbers generation)**.

Application note

In accordance with [JCAPI301], the keys are generated and diversified in class KeyBuilder (buildKey method); the related key class is HMACKey of javacard.security.

As mentioned in [JCAPI301] the key can be of any length, but it is strongly recommended that the key is not shorter than the byte length of the hash output used in the HMAC implementation. Keys with length greater than the hash block length are first hashed with the hash algorithm used for the HMAC implementation. As required, the implementation also supports an HMAC key length equal to the length of the supported hash algorithm block size.

**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 **[method SetKEY of DESKey class]** that meets the following: **[JCAPI301] standard**.

**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 **[method SetKEY of AESKey class]** that meets the following: **[JCAPI301] standard.**

**FCS_CKM.2/RSA Cryptographic key distribution**

**FCS_CKM.2.1/RSA** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[see application note]** that meets the following: **[JCAPI301] standard.**

Application note: the key distribution methods related to each key class are listed below:

| Key class | Key access methods |
|---|---|
| RSAPrivateKey | setExponent<br>setModulus |
| RSAPrivateCrtKey | setDP1<br>setDQ1<br>setP<br>setPQ<br>setQ |
| RSAPublicKey | setExponent<br>setModulus |

**FCS_CKM.2/ECDSA Cryptographic key distribution**

**FCS_CKM.2.1/ECDSA** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[see application note]** that meets the following: **[JCAPI301] standard.**

Application note: the key distribution methods related to each key class are listed below:

| Key class | Key distribution method |
|---|---|
| ECPrivateKey | setS |
| ECPublicKey | setW |

**FCS_CKM.2/HMAC Cryptographic key distribution**

**FCS_CKM.2.1/HMAC** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[method SetKEY of HMACKey class]** that meets the following: **[JCAPI301] standard**.

**FCS_CKM.3/TDES Cryptographic key access**

**FCS_CKM.3.1/TDES** The TSF shall perform **3DES key access** in accordance with a specified cryptographic key access method **[method getKey of the DESKey class]** that meets the following: **[JCAPI301] standard**.

### FCS_CKM.3/AES Cryptographic key access

**FCS_CKM.3.1/AES** The TSF shall perform **AES key access** in accordance with a specified cryptographic key access method **[method getKey of the AESKey class]** that meets the following: **[JCAPI301] standard**.

### FCS_CKM.3/RSA Cryptographic key access

**FCS_CKM.3.1/RSA** The TSF shall perform **RSA key access** in accordance with a specified cryptographic key access method **[see application note]** that meets the following: **[JCAPI301] standard**.

Application note: the key access methods related to each key class are listed below:

| Key class | Key access methods |
|---|---|
| RSAPrivateKey | getExponent<br>getModulus |
| RSAPrivateCrtKey | getDP1<br>getDQ1<br>getP<br>getPQ<br>getQ |
| RSAPublicKey | getExponent<br>getModulus |

### FCS_CKM.3/ECDSA Cryptographic key access

**FCS_CKM.3.1/ECDSA** The TSF shall perform **ECDSA key access** in accordance with a specified cryptographic key access method **[see application note]** that meets the following: **[JCAPI301] standard**.

Application note: the key access methods related to each key class are listed below:

| Key class | Key access methods |
|---|---|
| ECPrivateKey | getS |
| ECPublicKey | getW |

### FCS_CKM.3/HMAC Cryptographic key access

**FCS_CKM.3.1/HMAC** The TSF shall perform **HMAC key access** in accordance with a specified cryptographic key access method **[method getKey of the HMACKey class]** that meets the following: **[JCAPI301] standard**.

### FCS_CKM.4 Cryptographic key destruction

**FCS_CKM.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[see application note]** that meets the following: **[JCAPI301] standard**.

Application note: the keys are reset as specified in [JCAPI301] Key class, with the method clearKey. Any access to a cleared key for ciphering or signing shall throw an exception.

### FCS_COP.1/TDES_CIPHER Cryptographic operation

**FCS_COP.1.1/TDES_CIPHER** The TSF shall perform **[encryption and decryption of applet instance's data]** in accordance with a specified cryptographic algorithm **[Triple DES 2 Keys or Triple DES 3 Keys with cipher modes and padding schemes mentioned in the application note below]** and cryptographic key sizes **[112 bits for TDES 2 Keys, 168 bits for TDES 3 Keys]** that meet the following: **[FIPS PUB 46-3, FIPS PUB 81, ISO/IEC 9797-1, PKCS#5]**.

Application note: the following TDES ciphers from [JCAPI301] are implemented:

| Mode | Padding scheme | Field name in [JCAPI301] Cipher class |
|---|---|---|
| CBC | None (no padding) | ALG_DES_CBC_NOPAD |
| CBC | ISO9797 method 1 | ALG_DES_CBC_ISO9797_M1 |
| CBC | ISO9797 method 2 | ALG_DES_CBC_ISO9797_M2 |
| CBC | PKCS#5 | ALG_DES_CBC_PKCS5 |
| ECB | None (no padding) | ALG_DES_ECB_NOPAD |
| ECB | ISO9797 method 1 | ALG_DES_ECB_ISO9797_M1 |
| ECB | ISO9797 method 2 | ALG_DES_ECB_ISO9797_M2 |
| ECB | PKCS#5 | ALG_DES_ECB_PKCS5 |

### FCS_COP.1/TDES_MAC Cryptographic operation

**FCS_COP.1.1/TDES_MAC** The TSF shall perform **[MAC computation of applet instance's data]** in accordance with a specified cryptographic algorithm **[MAC algorithms with padding schemes mentioned in the application note below]** and cryptographic key sizes **[112 bits for TDES 2 Keys, 168 bits for TDES 3 Keys]** that meet the following: **[FIPS PUB 46-3, FIPS PUB 81, ISO/IEC 9797-1, PKCS#5]**.

Application note: the following TDES MACs from [JCAPI301] are implemented:

| MAC length | MAC algorithm | Padding scheme | Field name in [JCAPI301] Signature class |
|---|---|---|---|
| 4 bytes | ISO9797-1 MAC algorithm 3 | ISO9797-1 method 2 | ALG_DES_MAC4_ISO9797_1_M2_ALG3 |
| 4 bytes | 3DES in outer CBC mode | ISO9797-1 method 1 | ALG_DES_MAC4_ISO9797_M1 |
| 4 bytes | 3DES in outer CBC mode | ISO9797-1 method 2 | ALG_DES_MAC4_ISO9797_M2 |
| 4 bytes | 3DES in outer CBC mode | PKCS#5 | ALG_DES_MAC4_PKCS5 |
| 4 bytes | 3DES in outer CBC mode | None | ALG_DES_MAC4_NOPAD |
| 8 bytes | ISO9797-1 MAC algorithm 3 | ISO9797-1 method 2 | ALG_DES_MAC8_ISO9797_1_M2_ALG3 |
| 8 bytes | 3DES in outer CBC mode | ISO9797-1 method 1 | ALG_DES_MAC8_ISO9797_M1 |
| 8 bytes | 3DES in outer CBC mode | ISO9797-1 method 2 | ALG_DES_MAC8_ISO9797_M2 |
| 8 bytes | 3DES in outer CBC mode | PKCS#5 | ALG_DES_MAC8_PKCS5 |
| 8 bytes | 3DES in outer CBC mode | None | ALG_DES_MAC8_NOPAD |

### FCS_COP.1/AES_CIPHER Cryptographic operation

**FCS_COP.1.1/AES_CIPHER** The TSF shall perform **[encryption and decryption of applet instance's data]** in accordance with a specified cryptographic algorithm **[AES with cipher modes and padding schemes mentioned in the application note below]** and cryptographic key sizes **[128, 192 and 256 bits]** that meet the following: **[FIPS PUB 197, NIST SP800-38A , ISO/IEC 9797-1]**.

Application note: the following AES ciphers from [JCAPI301] are implemented:

| Mode | Padding scheme | Field name in [JCAPI301] Cipher class |
|---|---|---|
| CBC | None (no padding) | ALG_AES_BLOCK_128_CBC_NOPAD |
| CBC | ISO9797 method 1 | ALG_AES_CBC_ISO9797_M1 |
| CBC | ISO9797 method 2 | ALG_AES_CBC_ISO9797_M2 |
| CBC | PKCS#5 | ALG_AES_CBC_PKCS5 |
| ECB | None (no padding) | ALG_AES_BLOCK_128_ECB_NOPAD |

| | | |
|---|---|---|
| ECB | ISO9797 method 1 | ALG_AES_ECB_ISO9797_M1 |
| ECB | ISO9797 method 2 | ALG_AES_ECB_ISO9797_M2 |
| ECB | PKCS#5 | ALG_AES_ECB_PKCS5 |

**FCS_COP.1/AES_MAC Cryptographic operation**

**FCS_COP.1.1/AES_MAC** The TSF shall perform **[MAC computation of applet instance's data]** in accordance with a specified cryptographic algorithm **[MAC algorithms mentioned in the application note below]** and cryptographic key sizes **[128, 192 and 256 bits]** that meet the following: **[FIPS PUB 197, NIST SP800-38A]**.

Application note: the following AES MACs from [JCAPI301] are implemented:

| MAC length | MAC algorithm | Padding scheme | Field name in [JCAPI301] Signature class |
|---|---|---|---|
| 16 bytes | AES in CBC mode, block size 128 bits | None | ALG_AES_MAC_128_NOPAD |
| 24 bytes | AES in CBC mode, block size 192 bits | None | ALG_AES_MAC_192_NOPAD |
| 32 bytes | AES in CBC mode, block size 256 bits | None | ALG_AES_MAC_256_NOPAD |

**FCS_COP.1/RSA_SIGN Cryptographic operation**

**FCS_COP.1.1/RSA_SIGN** The TSF shall perform **[signature generation and signature verification of applet instance's data]** in accordance with a specified cryptographic algorithm **[RSA Standard and RSA CRT with hash algorithms and padding schemes mentioned in the application note below]** and cryptographic key sizes **[512 to 2048 bits by steps of 32 bits]** that meet the following: **[PKCS #1 Version 2.1, PKCS#1-PSS (IEEE 1363-2000), ISO/IEC 9796-2 and RFC2409]**.

Application note: the following RSA signatures from [JCAPI301] are implemented:

| Hash algorithm | Padding scheme | Field name in [JCAPI301] Signature class |
|---|---|---|
| MD5 | PKCS#1 | ALG_RSA_MD5_PKCS1 |
| MD5 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_MD5_PKCS1_PSS |
| MD5 | RFC2409 | ALG_RSA_MD5_RFC2409 |
| SHA224 | PKCS#1 | ALG_RSA_SHA_224_PKCS1 |
| SHA224 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_SHA_224_PKCS1_PSS |
| SHA256 | PKCS#1 | ALG_RSA_SHA_256_PKCS1 |
| SHA256 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_SHA_256_PKCS1_PSS |
| SHA384 | PKCS#1 | ALG_RSA_SHA_384_PKCS1 |
| SHA384 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_SHA_384_PKCS1_PSS |
| SHA512 | PKCS#1 | ALG_RSA_SHA_512_PKCS1 |
| SHA512 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_SHA_512_PKCS1_PSS |
| SHA1 | ISO 9796-2 | ALG_RSA_SHA_ISO9796 |
| SHA1 | PKCS#1 | ALG_RSA_SHA_PKCS1 |
| SHA1 | PKCS#1-PSS scheme (IEEE 1363-2000) | ALG_RSA_SHA_PKCS1_PSS |
| SHA1 | RFC2409 | ALG_RSA_SHA_RFC2409 |

**FCS_COP.1/RSA_CIPHER Cryptographic operation**

**FCS_COP.1.1/RSA_CIPHER** The TSF shall perform **[encryption and decryption of applet instance's data]** in accordance with a specified cryptographic algorithm **[RSA Standard and RSA CRT as mentioned in the application note below]** and cryptographic key sizes **[512 to 2048 bits by steps of 32 bits]** that meet the following: **[PKCS #1 Version 2.1, PKCS#1-OAEP scheme (IEEE 1363-2000)]**.

Application note: the following RSA ciphers from [JCAPI301] are implemented:

| [JCAPI301] class | Implemented algorithms |
|---|---|
| Cipher | ALG_RSA_NOPAD |
| | ALG_RSA_PKCS1 |
| | ALG_RSA_PKCS1_OAEP |

**FCS_COP.1/ECDSA_SIGN Cryptographic operation**

**FCS_COP.1.1/ECDSA_SIGN** The TSF shall perform **[signature generation and signature verification of applet instance's data]** in accordance with a specified cryptographic algorithm **[ECDSA as mentioned in the application note below]** and cryptographic key sizes **[P ranging from 192 to 521 bits]** that meet the following: **[FIPS PUB 186-2]**.

Application note: the following ECDSA signatures from [JCAPI301] are implemented:

| Hash algorithm | Field name in [JCAPI301] Signature class |
|---|---|
| SHA1 | ALG_ECDSA_SHA |
| SHA224 | ALG_ECDSA_SHA_224 |
| SHA256 | ALG_ECDSA_SHA_256 |
| SHA384 | ALG_ECDSA_SHA_384 |
| SHA512 | ALG_ECDSA_SHA_512 |

**FCS_COP.1/ECDH Cryptographic operation**

**FCS_COP.1.1/ECDH** The TSF shall perform **[Secret Key Agreement]** in accordance with a specified cryptographic algorithm **Elliptic Curve Diffie-Hellman (ECDH)** and cryptographic key sizes **[P ranging from 192 to 521 bits]** that meet the following: **[IEEE P1363]**.

Application note: the secret keys are derived using the KeyAgreement class (generateSecret method) of javacard.security.

**FCS_COP.1/Hash Cryptographic operation**

**FCS_COP.1.1/Hash** The TSF shall perform **[computation of a hash value for applet instance's data]** in accordance with a specified cryptographic algorithm **[see application note]** and cryptographic key sizes **[None]** that meet the following: **[see application note]**.

Application note: the following hash algorithms from [JCAPI301] are implemented:

| Hash algorithm | Field name in [JCAPI301] MessageDigest class | Related Standard |
|---|---|---|
| MD5 | ALG_MD5 | |
| SHA1 | ALG_SHA | FIPS 180-3 |
| SHA224 | ALG_SHA_224 | FIPS 180-3 |
| SHA256 | ALG_SHA_256 | FIPS 180-3 |
| SHA384 | ALG_SHA_384 | FIPS 180-3 |
| SHA512 | ALG_SHA_512 | FIPS 180-3 |

**FCS_COP.1/HMAC Cryptographic operation**

**FCS_COP.1.1/HMAC** The TSF shall perform **[computation of a HMAC value for applet instance's data]** in accordance with a specified cryptographic algorithm **[HMAC with hash algorithms mentioned in the application note below]** and cryptographic key sizes **[see application note]** that meet the following: **[rfc2104]**.

Application note

The following HMAC algorithms from [JCAPI301] are implemented:

| Hash algorithm used in HMAC computation | Field name in [JCAPI301] Signature class |
|---|---|
| MD5 | ALG_HMAC_MD5 |
| SHA1 | ALG_HMAC_SHA1 |
| SHA256 | ALG_HMAC_SHA_256 |
| SHA384 | ALG_HMAC_SHA_384 |
| SHA512 | ALG_HMAC_SHA_512 |

As mentioned in [JCAPI301] the key can be of any length, but it is strongly recommended that the key is not shorter than the byte length of the hash output used in the HMAC implementation. Keys with length greater than the hash block length are first hashed with the hash algorithm used for the HMAC implementation. As required, the implementation also supports an HMAC key length equal to the length of the supported hash algorithm block size.

**FCS_COP.1/CRC Cryptographic operation**

**FCS_COP.1.1/CRC** The TSF shall perform **[Computation of checksum of applet instance's data]** in accordance with a specified cryptographic algorithm **[CRC16 or CRC32]** and cryptographic key sizes **[none]** that meet the following: **[ISO/IEC 3309]**.

Application note: the related algorithms in [JCAPI301] are ALG_ISO3309_CRC16 and ALG_ISO3309_CRC32 (class Checksum of javacard.security).

**FCS_RND.1 Random Number Generation**

**FCS_RND.1.1** The TSF shall provide a mechanism to generate random numbers that meet **the quality level specified in NIST SP800-90**.

**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**.

Application note: the events that provoke the de-allocation of a transient object are described in [JCRE22], §5.1.

**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**.

Application note: the allocation of a resource to the APDU buffer is typically performed as the result of a call to the process() method of an applet.

**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**.

Application note: a resource is allocated to the bArray object when a call to an applet's install() method is performed. There is no conflict with FDP_ROL.1 here because of the bounds on the rollback

mechanism (FDP_ROL.1.2/FIREWALL): the scope of the rollback does not extend outside the execution of the install() method, and the de-allocation occurs precisely right after the return of it.

| 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)**.

Application note: the javacard.security & javacardx.crypto packages do provide secure interfaces to the cryptographic buffer in a transparent way. See javacard.security.KeyBuilder and Key interface of [JCAPI301].

| 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. Moreover in [JCRE301] §3.6.1, Transient data of CLEAR_ON_DESELECT objects associated with each applet instance that was active on a logical channel over the contactless I/O interface and that does not have an applet instance from the same package active on any logical channel over the contacted I/O interface, is reset to the default value.

| 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 **operations OP.JAVA and OP.CREATE** on the **object O.JAVAOBJECT**.

**FDP_ROL.1.2/FIREWALL** The TSF shall permit operations to be rolled back within the **scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in [JCRE22], §7.7, within the bounds of the Commit Capacity ([JCRE22], §7.8), and those described in [JCAPI22]**.
Application note: transactions are a service offered by the APIs to applets. It is also used by some APIs to guarantee the atomicity of some operation. This mechanism is either implemented in Java Card platform or relies on the transaction mechanism offered by the underlying platform. Some operations of the API are not conditionally updated, as documented in [JCAPI22] (see for instance, PIN-blocking, PIN-checking, update of Transient objects).

Card Security Management

### FAU_ARP.1 Security alarms

**FAU_ARP.1.1** The TSF shall take **one of the following actions: throw an exception, lock the card session, reinitialize the Java Card System and its data,** upon detection of a potential security violation.

Refinement: the "potential security violation" stands for one of the following events:

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card tearing (unexpected removal of the Card out of the CAD) and power failure,
- abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI22] and ([JCRE22], §7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow.

### FDP_SDI.2 Stored data integrity monitoring and action

**FDP_SDI.2.1** The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: **integrityCheckData**.

**FDP_SDI.2.2** Upon detection of a data integrity error, the TSF shall **increase a counter of integrity error event and mute the card if counter is greater than max value**.

Application note: the following data persistently stored by TOE have an integrity check data security attribute:

- Key (i.e. objects instance of classes implemented the interface Key)
- PIN (objects instance of class OwnerPin)
- Package.

### FPR_UNO.1 Unobservability

**FPR_UNO.1.1** The TSF shall ensure that **[any user]** are unable to observe the operation **[read, write, cryptographic operations]** on **[PIN, Key]** by **[any other user or subject]**.

Application note: although it is not required in [JCRE301] specifications, the non-observability of operations on sensitive information such as keys appears as impossible to circumvent in the smart card world.

### 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**.

Application note: the Java Card RE Context is the Current context when the Java Card VM begins running after a card reset ([JCRE22], §6.2.3) or after a proximity card (PICC) activation sequence ([JCRE222]). Behavior of the TOE on power loss and reset is described in [JCRE22], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [JCRE222], §3.6.1.

### 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, the bytecode and its data arguments** when shared between the TSF and another trusted IT product.

**FPT_TDC.1.2** The TSF shall use
- **the rules defined in [JCVM301] specification,**
- **the API tokens defined in the export files of reference implementation,**

When interpreting the TSF data from another trusted IT product.

Application note: concerning the interpretation of data between the TOE and the underlying Java Card platform, it is assumed that the TOE is developed consistently with the SCP functions, including memory management, I/O functions and cryptographic functions.

---

**AID management**

| 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:
- **Package AID,**
- **Applet's version number,**
- **Registered applet AID,**
- **Applet Selection Status ([JCVM22], §6.5)**.

Refinement: "Individual users" stand for applets.

| FIA_UID.2/AID | User identification before any action |
|---|---|

**FIA_UID.2.1/AID** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application note:
- By users here it must be understood the ones associated to the packages (or applets) that 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.
- The role Java Card RE defined in FMT_SMR.1 is attached to an IT security function rather than to a "user" of the CC terminology. The Java Card RE does not "identify" itself to the TOE, but it is part of it.

| FIA_USB.1/AID | User-subject binding |
|---|---|

**FIA_USB.1.1/AID** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **Package AID**, **active context**.

**FIA_USB.1.2/AID** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **Package AID are defined with associated value during loading and with context identifier**.

**FIA_USB.1.3/AID** 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: the user is the applet and the subject is the S.PACKAGE. The subject security attribute "Context" shall hold the user security attribute "package AID".

| 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' AIDs** to **the JCRE**.

Application note:

- The installer and the Java Card RE manage other TSF data such as the applet life cycle or CAP files, but this management is implementation specific. Objects in the Java programming language may also try to query AIDs of installed applets through the lookupAID(...) API method.
- The installer, applet deletion manager or even the card manager may be granted the right to modify the list of registered applets' AIDs in specific implementations (possibly needed for installation and deletion; see #.DELETION and #.INSTALL).

**FMT_MTD.3/JCRE Secure TSF data**

**FMT_MTD.3.1/JCRE** The TSF shall ensure that only secure values are accepted for **the registered applets' AIDs**.

**InstG Security Functional Requirements**

This group consists of the SFRs related to the installation of the applets, which addresses security aspects outside the runtime. The installation of applets is a critical phase, which lies partially out of the boundaries of the firewall, and therefore requires specific treatment. In this PP, loading a package or installing an applet modeled as importation of user data (that is, user application's data) with its security attributes (such as the parameters of the applet used in the firewall rules).

Application note: patch installation is an extension of applet installation defined in GP as a patch is managed as a JavaCard Package and loaded as a standard executable load file and registered with specific attributes handled with GemActivate.

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

**FDP_ITC.2.1/Installer** The TSF shall enforce the **PACKAGE LOADING information flow control SFP** when importing user data, controlled under the SFP, from outside of the TOE.

**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 TOE:
**Package 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 dependent package is lesser than or equal to the major (minor) Version attribute associated to the resident package ([JCVM22], §4.5.2).**

Application note, FDP_ITC.2.1/Installer:
The most common importation of user data is package loading and applet installation on the behalf of the installer. Security attributes consist of the shareable flag of the class component, AID and version numbers of the package, maximal operand stack size and number of local variables for each method, and export and import components (accessibility).

Application note, FDP_ITC.2.3/Installer:
The format of the CAP file is precisely defined in [JCVM301] specifications; it contains the user data (like applet's code and data) and the security attributes altogether. Therefore there is no association to be carried out elsewhere.

Application note, FDP_ITC.2.4/Installer:
Each package contains a package Version attribute, which is a pair of major and minor version numbers ([JCVM22], §4.5). With the AID, it describes the package defined in the CAP file. When an export file is used during preparation of a CAP file, the versions numbers and AIDs indicated in the export file are recorded in the CAP files ([JCVM22], §4.5.2): the dependent packages Versions and AIDs attributes allow the retrieval of these identifications. Implementation-dependent checks may occur on a case-by-case basis to indicate that package files are binary compatible. However, package files do have "package Version Numbers" ([JCVM22]) used to indicate binary compatibility or incompatibility between successive implementations of a package, which obviously directly concern this requirement.

Application note, FDP_ITC.2.5/Installer:
- 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 intent of this rule is to ensure the binary compatibility of the package with those already on the card ([JCVM22], §4.4).
- The installation (the invocation of an applet's install method by the installer) is implementation dependent ([JCRE22], §11.2).
- Other rules governing the installation of an applet, that is, its registration to make it SELECTable by giving it a unique AID, are also implementation dependent (see, for example, [JCRE22], §11).

**FMT_SMR.1/Installer Security roles**

**FMT_SMR.1.1/Installer** The TSF shall maintain the roles**: Installer**.

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

**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] §11.1.4**.

Application note:
- The TOE may provide additional feedback information to the card manager in case of potential security violations (see FAU_ARP.1).
- Patch is also loaded and installed using such rules as a standard executable load file with specific attributes handled with GemActivate.

**FPT_RCV.3/Installer Automated recovery without undue loss**

**FPT_RCV.3.1/Installer** When automated recovery from **a failure or service discontinuity** 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/Installer** For **[detection of a potential loss of integrity during the transmission of an Executable Load File to the card, abortion of the installation process of an Executable Load File, or any fatal error occurred during the linking of an Executable Load File to the Executable Files already installed on the card]**, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

**FPT_RCV.3.3/Installer** The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **[the loss of the Executable Load File being installed]** for loss of TSF data or objects under the control of the TSF.

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

Application notes:

Patch is also loaded and installed using such rules as a standard executable load file with specific attributes handled with GemActivate.

FPT_RCV.3.1/Installer: this element is not within the scope of the Java Card specification, which only mandates the behavior of the Java Card System in good working order. Further details on the "maintenance mode" shall be provided in specific implementations. The following is an excerpt from [CC-2], p298: In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorized users should be allowed access to this mode but the real details of who can access this mode is a function of FMT: Security management. If FMT: Security management does not put any controls on who can access this mode, then it may be acceptable to allow any user to restore the system if the TOE enters such a state. However, in practice, this is probably not desirable as the user restoring the system has an opportunity to configure the TOE in such a way as to violate the SFRs.

FPT_RCV.3.2/Installer: should the installer fail during loading/installation of a package/applet, it has to revert to a "consistent and secure state". The Java Card RE has some clean up duties as well; see [JCRE22], §11.1.5 for possible scenarios. Precise behavior is left to implementers. This component shall include among the listed failures the deletion of a package/applet. See ([JCRE22], 11.3.4) for possible scenarios. Precise behavior is left to implementers. Other events such as the unexpected tearing of the card, power loss, and so on, are partially handled by the underlying hardware platform (see [PP0035]) and, from the TOE's side, by events "that clear transient objects" and transactional features. See FPT_FLS.1.1, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ABORT and FDP_ROL.1/FIREWALL.

FPT_RCV.3.3/Installer: the quantification is implementation dependent, but some facts can be recalled here. First, the SCP ensures the atomicity of updates for fields and objects, and a power-failure during a transaction or the normal runtime does not create the loss of otherwise-permanent data, in the sense that memory on a smart card is essentially persistent with this respect (EEPROM). Data stored on the RAM and subject to such failure is intended to have a limited lifetime anyway (runtime data on the stack, transient objects' contents). According to this, the loss of data within the TSF scope should be limited to the same restrictions of the transaction mechanism.

### ADELG Security Functional Requirements

This group consists of the SFRs related to the deletion of applets and/or packages, enforcing the applet deletion manager (ADEL) policy on security aspects outside the runtime. Deletion is a critical operation and therefore requires specific treatment.

Application note: patch deletion is an extension of applet/package deletion defined in GP as a patch is managed as a JavaCard Package and registered with specific attributes handled with GemActivate.

#### FDP_ACC.2/ADEL Complete access control

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

Refinement: the operations involved in the policy are: OP.DELETE_APPLET, OP.DELETE_PCKG, and OP.DELETE_PCKG_APPLET.

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

## FDP_ACF.1/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:

| Subject / Object | Attributes |
|---|---|
| S.JCVM | Active Applets |
| S.JCRE | Selected Applet Context, Registered Applets, Resident Packages |
| O.CODE_PKG | Package AID, Dependent Package AID, Static References |
| O.APPLET | Applet Selection Status |
| O.JAVAOBJECT | Owner |

**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:

- **In the context of this policy, an object O is reachable if and only one of the following conditions hold:**
  1) **the owner of O is a registered applet instance A (O is reachable from A),**
  2) **a static field of a resident package P contains a reference to O (O is reachable from P),**
  3) **there exists an object O' that is reachable according to either 1) or 2) 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): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,**
    1) **S.ADEL is currently selected,**
    2) **there is no instance in the context of O.APPLET that is active in any logical channel 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.**
  - **R.JAVA.15 ([JCRE22], §11.3.4.1, Multiple Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,**
    1) **S.ADEL is currently selected,**
    2) **there is no instance of any of the O.APPLET being deleted that is active in any logical channel 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.**
  - **R.JAVA.16 ([JCRE22], §11.3.4.2, Applet/Library Package Deletion): S.ADEL may perform OP.DELETE_PCKG upon an O.CODE_PKG only if,**
    1) **S.ADEL is currently selected,**
    2) **no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG that is an instance of a class that belongs to O.CODE_PKG, exists on the card and**
    3) **there is no resident package on the card that depends on O.CODE_PKG.**
  - **R.JAVA.17 ([JCRE22], §11.3.4.3, Applet Package and Contained Instances Deletion): S.ADEL may perform OP.DELETE_PCKG_APPLET upon an O.CODE_PKG only if,**
    1) **S.ADEL is currently selected,**
    2) **no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG, which is an instance of a class that belongs to O.CODE_PKG exists on the card,**
    3) **there is no package loaded on the card that depends on O.CODE_PKG, and**
    4) **for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel 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.**

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

**FDP_ACF.1.4/ADEL** **[Editorially Refined]** The TSF shall explicitly deny access of **any subject but S.ADEL to O.CODE_PKG or O.APPLET for the purpose of deleting them from the card**.

Application note, FDP_ACF.1.2/ADEL: this policy introduces the notion of reachability, which provides a general means to describe objects that are referenced from a certain applet instance or package. S.ADEL calls the "uninstall" method of the applet instance to be deleted, if implemented by the applet, to inform it of the deletion request. The order in which these calls and the dependencies checks are performed are out of the scope of this security target.

**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: deleted freed resources (both code and data) may be reused, depending on the way they were deleted (logically or physically). 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.

**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 **Registered Applets and Resident Packages** to **the Java Card RE**.

Application note: patch deletion is an extension of applet/package deletion defined in GP as a patch is managed as a JavaCard Package and registered with specific attributes.

**FMT_MSA.3/ADEL Static attribute initialization**

**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 role(s): none,** to specify alternative initial values to override the default values when an object or information is created.

Application note: patch deletion is an extension of applet/package deletion defined in GP as a patch is managed as a JavaCard Package and registered with specific attributes.

**FMT_SMF.1/ADEL Specification of Management Functions**

**FMT_SMF.1.1/ADEL** The TSF shall be capable of performing the following management functions: **modify the list of registered applets' AIDs and the Resident Packages**.

Application note: the modification of the Active Applets security attribute should be performed in accordance with the rules given in [JCRE22], §4.

**FMT_SMR.1/ADEL Security roles**

**FMT_SMR.1.1/ADEL** The TSF shall maintain the roles**: applet deletion manager**.

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

**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**.

Application note:

- The TOE may provide additional feedback information to the card manager in case of a potential security violation (see FAU_ARP.1).
- The Package/applet instance deletion must be atomic. The "secure state" referred to in the requirement must comply with Java Card specification ([JCRE22], §11.3.4.)

Application note: patch deletion is an extension of applet/package deletion defined in GP as a patch is managed as a JavaCard Package and registered with specific attributes.

### ODELG Security Functional Requirements

The following requirements concern the object deletion mechanism. This mechanism is triggered by the applet that owns the deleted objects by invoking a specific API method.

#### 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()**

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].
- There is no conflict with FDP_ROL.1 here because of the bounds on the rollback mechanism: the execution of requestObjectDeletion() is not in the scope of the rollback because it must be performed in between APDU command processing, and therefore no transaction can be in progress.

#### 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**.

Application note: the TOE may provide additional feedback information to the card manager in case of potential security violation (see FAU_ARP.1).

### CarG Security Functional Requirements

This group includes requirements for preventing the installation of packages that have not been bytecode verified, or that have been modified after bytecode verification.

#### 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** **[Editorially Refined]** The TSF shall be able to relate the **identity** of the originator of the information, and the **application package contained in** the information to which the evidence applies.

**FCO_NRO.2.3/CM** The TSF shall provide a capability to verify the evidence of origin of information to **recipient** given **as assumption the key used is kept integer and confidential by origin**.

Application note, FCO_NRO.2.1/CM: upon reception of a new application package 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.

Application note, FCO_NRO.2.3/CM: the exact limitations on the evidence of origin are implementation dependent. In most of the implementations, the card manager performs an immediate verification of the origin of the package using an electronic signature mechanism, and no evidence is kept on the card for future verifications.

| 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.INSTALLER, S.BCV, S.CAD and I.APDU** and all operations that cause that information to flow to and from subjects covered by the SFP.

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

Application note:

- The subjects covered by this policy are those involved in the loading of an application package by the card through a potentially unsafe communication channel.
- The operations 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 an attacker. Moreover, an attacker may capture any message sent through the communication channel and send its own messages to the other subjects.
- The information controlled by the 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, as well as any control information used by the subjects in the communication protocol.

| 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: **[the Command Security Level defined for the messages that the card receives through the secure channel]**.

**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: **[assignment: the rules describing the communication protocol used by the CAD and the card for transmitting a new package]**.

**FDP_IFF.1.3/CM** The TSF shall enforce the **[possible security levels are: NO-SEC (clear text), C-AUTHENTICATED (authentication of the command's emitter), C-MAC (authentication of the emitter and integrity of the command), C-DEC (authentication of the emitter, integrity and confidentiality of the command)]**.

**FDP_IFF.1.4/CM** The TSF shall explicitly authorize an information flow based on the following rules: **[the SD may process:**

- **an (INITIALIZE-UPDATE) operation only if the key set specified in the command exist,**
- **an (EXTERNAL-AUTHENTICATE) operation if the following conditions are fulfilled:**
  1) **The cryptogram received from the off-card subject is equal to the cryptogram computed by the Security Domain.**
  2) **The MAC attached to the message has been generated using the CMAC session key and the current value of the ICV.**
- **a (GET-DATA) operation if the following condition are fulfilled:**
  1) **If the command security level is at least C-MAC,**

2) **The MAC attached to the message has been generated from the command using the C-MAC session key and the current value of the ICV.**

- **any received operation for any other command if the following conditions hold:**
  1) **The current security level is at least AUTHENTICATED.**
  2) **If the command security level is at least C-MAC, the MAC attached to the message has been generated from the clear-text command using the C-MAC session key and the current value of the ICV.**

**FDP_IFF.1.5/CM** The TSF shall explicitly deny an information flow based on the following rules: **[A Security Domain may always process a (SELECT) operation or a (Get DATA) operation at the security level NO-SEC]**.

Application note:

- FDP_IFF.1.1/CM: the security attributes used to enforce the PACKAGE LOADING SFP are implementation dependent. More precisely, they depend on the communication protocol enforced between the CAD and the card. For instance, some of the attributes that can be used are: (1) the keys used by the subjects to encrypt/decrypt their messages; (2) the number of pieces the application package has been split into in order to be sent to the card; (3) the ordinal of each piece in the decomposition of the package, etc. See for example Appendix D of [GP].
- FDP_IFF.1.2/CM: the precise set of rules to be enforced by the function is implementation dependent. The whole exchange of messages shall verify at least the following two rules: (1) the subject S.INSTALLER shall accept a message only if it comes from the subject S.CAD; (2) the subject S.INSTALLER shall accept an application package only if it has received without modification and in the right order all the APDUs sent by the subject S.CAD.

**FDP_UIT.1/CM Data exchange integrity**

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

**FDP_UIT.1.2/CM** **[Editorially Refined]** The TSF shall be able to determine on receipt of user data, whether **modification, deletion, insertion, replay of some of the pieces of the application sent by the CAD** has occurred.

Application note: modification errors should be understood as modification, substitution, unrecoverable ordering change of data and any other integrity error that may cause the application package to be installed on the card to be different from the one sent by the CAD.

**FIA_UID.1/CM Timing of identification**

**FIA_UID.1.1/CM** The TSF shall allow **selection of a security domain and execution of Card Manager** on behalf of the user to be performed before the user is identified.

**FIA_UID.1.2/CM** 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 list of TSF-mediated actions is implementation-dependent, but package installation requires the user to be identified. Here by user is meant the one(s) that in the Security Target shall be associated to the role(s) defined in the component FMT_SMR.1/CM.

**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 **[Card Life cycle, Security Level]** to **[Card Manager]**.

| FMT_MSA.3/CM | Static attribute initialization |
|---|---|

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

| FMT_SMF.1/CM | Specification of Management Functions |
|---|---|

**FMT_SMF.1.1/CM** The TSF shall be capable of performing the following management functions: **[modification of the Card life cycle inducing availability of management functions]**.

| FMT_SMR.1/CM | Security roles |
|---|---|

**FMT_SMR.1.1/CM** The TSF shall maintain the roles **[S.CAD, S.CARDMANAGER]**.

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

| FTP_ITC.1/CM | Inter-TSF trusted channel |
|---|---|

**FTP_ITC.1.1/CM** The TSF shall provide a communication channel between itself and another 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** **[Editorially Refined]** The TSF shall permit **the CAD placed in the card issuer secured environment** to initiate communication via the trusted channel.

**FTP_ITC.1.3/CM** The TSF shall initiate communication via the trusted channel for **loading/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.

## SCP Security Functional Requirements

This section states the security functional requirements for the Smart Card Platform.

### Operating System

This section presents those requirements of the Smart Card Platform group that concern the Operating System. Due to the enlargement of the evaluation scope, the requirements related to OS are now assigned to the TOE and no more to the environment. Other internal security mechanisms are not addressed by SFR but ADV_ARC activities.

| FPT_RCV.3/OS | Automated recovery without undue loss |
|---|---|

**FPT_RCV.3.1/OS** When automated recovery from **security policy violation** 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/OS** For **execution access to a memory zone reserved for TSF data, writing access to a memory zone reserved for TSF's code, and any segmentation fault performed by a Java Card applet**, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

**FPT_RCV.3.3/OS** The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **o the contents of Java Card static fields, instance fields, and array positions that fall under the scope of an open transaction; o the Java Card objects that were allocated into the scope of an open transaction;**

**o the contents of Java Card transient objects; o any possible Executable Load File being loaded when the failure occurred** for loss of TSF data or objects under the control of the TSF.

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

| **FPT_RCV.4/OS** | **Function recovery** |
|---|---|

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

---

**Integrated Circuit**

The section should contain the requirements of the Smart Card Platform group concerning the Integrated Circuit. Due to enlargement of the evaluation scope, the requirements related to IC are now assigned to the TOE and no more to the environment. Those requirements are fulfilled in [ST_ST33G1M2] and are covered by the IC certificate reused in the composite evaluation process. There are not repeated here.

They mainly concern protecting the smart card's chip against physical tampering, preventing the disclosure of information when it is transferred from different physical parts of the chip, providing the basic DES operation, keeping a secure state when a malfunction is detected and providing an independent security domain for the hardware.

### *9.1.3. Supplementary Security Functional Requirements*

**Secure API - Security Functional Requirements**

| **FPT_FLS.1/SecureAPI** | **Failure with preservation of secure state** |
|---|---|

**FPT_FLS.1.1/SecureAPI** The TSF shall preserve a secure state when the following types of failures occur: **the application fails to perform a specific execution flow control protected by the Secure API**.

| **FPT_ITT.1/SecureAPI** | **Basic internal TSF data transfer protection** |
|---|---|

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

| **FPR_UNO.1/SecureAPI** | **Unobservability** |
|---|---|

**FPR_UNO.1.1/SecureAPI** The TSF shall ensure that **external attackers** are unable to observe the operation **as sensitive comparison or copy** on **sensitive objects defined** by **the application using the Secure API**.

**GemActivate - Security Functional Requirements**

Application note: Rules for GemActivate are also applicable for patch. Activation of patch follow rules defined for optional platform service handled with GemActivate.

| **FMT_SMR.1/GemActivate** | **Security roles** |
|---|---|

**FMT_SMR.1.1/GemActivate** The TSF shall maintain the roles **[GemActivate Administrator]**.

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

| FMT_SMF.1/GemActivate | Specification of Management Functions |
|---|---|

**FMT_SMF.1.1/GemActivate** The TSF shall be capable of performing the following management functions: **activation of optional platform service**.

| FMT_MOF.1/GemActivate | Management of security functions behavior |
|---|---|

**FMT_MOF.1.1/GemActivate** The TSF shall restrict the ability to **disable and enable** the functions **activation or inhibition of optional platform services as: cryptographic algorithm, package, applet instance, scalability (extension of available NVM) and NFC interface (SWP, HCI gate)** to **GemActivate Administrator, MNO**.

Application note: activation and inhibition of patch follow the same rules as optional platform services.

| FMT_MSA.1/GemActivate | Management of security attributes |
|---|---|

**FMT_MSA.1.1/GemActivate** The TSF shall enforce the **GemActivate access control SFP** to restrict the ability to **modify** the security attributes **state (deactivated, activated, inhibited) of optional platform service** to **GemActivate Administrator under control of MNO**.

Application note: Patchability Activation Status is one of the optional platform services allowing patch activation. It is set by default to activated allowing activation of patch.

| FMT_MTD.1/GemActivate | Management of TSF data |
|---|---|

**FMT_MTD.1.1/GemActivate** The TSF shall restrict the ability to **query** the **[List of deactivated/activated/ inhibited optional platform services]** to **[GemActivate Administrator and MNO]**.

Application note: The list of patch package, patch and associated activation status is one of the available query.

| FDP_ACC.1/GemActivate_DAP | Subset access control |
|---|---|

**FDP_ACC.1.1/GemActivate_DAP** The TSF shall enforce the **GemActivate Security Domain access control policy** on:

- **Subjects: S.INSTALLER, S.GEMACTIVATE and S.SD**
- **Objects: GemActivate DAP Block and Load File**
- **Operations: Load and Install GlobalPlatform's APDU commands.**

Application note: Attributes of patch follow the same rules as optional platform services.

| FDP_ACF.1/ GemActivate_DAP | Security attribute based access control |
|---|---|

**FDP_ACF.1.1/GemActivate_DAP** The TSF shall enforce the **GemActivate Security Domain access control policy** to objects based on the following:

- **Subjects:**
  - **S.INSTALLER, defined in [PP-JCS] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [GP]);**
  - **S.GEMACTIVATE, responsible to compute DAP on load package using GemActivate key and to compare computed DAP with received DAP to authorize loading and linking with restricted packages.**
  - **S.SD receiving the Card Content Management commands (through Load and Install APDUs)**
- **Objects:**

  - **The GemActivate DAP Block, in case of application loading, referencing restricted packages (in particular the DESFIRE API);**
  - **The Load File or Executable File, in case of application loading, installation with a set of intended privileges and its targeted associated SD AID.**
- **The following security attributes:**
  - **The CardState attribute, is the current state in the life cycle of the card, which may be either OP_READY, INITIALIZED, SECURED, CARD_LOCKED, or TERMINATED.**
  - **The Restricted Linking attribute, is a flag authorizing link with restricted package which may be either AUTHORIZED or BLOCKED.**
  - **The Registered Applications attribute specifies the Executable Files and application instances that have been installed on the card so far and their dependencies.**

**FDP_ACF.1.2/GemActivate_DAP** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **Runtime behavior rules defined by GlobalPlatform for:**

- **loading (Section 9.3.5 of [GP]);**
- **installation (Section 9.3.6 of [GP]);**

**Where DAP verification is done using GemActivate key by GemActivate Administrator.**

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

**Rule GA-1: A loading and linking request for a package referencing restricted packages may be accepted only if the APDU command specifying the request contains a DAP well-formed according to [GP22] and its verification using GemActivate key by GemActivate Administrator is successful.**

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

- **Rule GA-2: When a loading and linking request for a package referencing restricted packages fails, package is not installed and associated NVM is recovered.**
- **Rule GA-3: When at least one of the rules for loading defined by GlobalPlatform [GP22] does not hold.**

Application note: Attributes of patch follow the same rules as optional platform services.

**FMT_MSA.3/GemActivate_DAP Static attribute initialization**

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

**FMT_MSA.3.2/GemActivate_DAP** The TSF shall allow **the GemActivate Administrator** to specify alternative initial values to override the default values when an object or information is created.

Application note:

- The Restricted Linking attribute for loading package with restricted import is set to Blocked by default. After a DAP verification, it is set to Authorized for the current loading and reset after successful or unsuccessful loading.
- Keyset and Key used for DAP verification are under control of GemActivate and key is imported securely in Personalization phase.

**FIA_ATD.1/AID_Patch User attribute definition**

**FIA_ATD.1.1/AID_Patch** The TSF shall maintain the following list of security attributes belonging to individual users:

- **Patch Package AID for each Patch package,**
- **Patch ID, Patch Activation Status for each patch.**

Refinement: "Individual users" stand for patch.

## 9.2. SECURITY ASSURANCE REQUIREMENTS

This ST is based on the EAL4 assurance package augmented with the components AVA_VAN.5 and ALC_DVS.2.

## 9.3. SECURITY REQUIREMENTS RATIONALE

### *9.3.1. TOE security objectives coverage – Mapping table*

| | O.CARD-MANAGEMENT | O.DOMAIN-RIGHTS | O.APPLI-AUTH | O.COMM_AUTH | O.COMM_INTEGRITY | O.COMM_CONFIDENTIALITY | O.SCP-SUPPORT | O.SID | O.FIREWALL | O.GLOBAL_ARRAYS_CONFID | O.GLOBAL_ARRAYS_INTEG | O.NATIVE | O.OPERATE | O.REALLOCATION | O.RESOURCES | O.ALARM | O.CIPHER | O.KEY-MNGT | O.PIN-MNGT | O.TRANSACTION | O.OBJ-DELETION | O.DELETION | O.LOAD | O.INSTALL | O.SCP.IC | O.SCP.RECOVERY | O.Secure_API | O.RND | O.JCAPI-Services | O.REMOTE_SERVICE_AUDIT | O.REMOTE_SERVICE_ACTIVATION | O.Secure_Load_ACode | O.Secure_AC_Activation | O.TOE_Identification |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FDP_UIT.1/CCM | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | |
| FDP_ROL.1/CCM | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X |
| FDP_ITC.2/CCM | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X |
| FPT_FLS.1/CCM | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | |
| FCS_COP.1/DAP | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FDP_ACC.1/SD | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FDP_ACF.1/SD | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_MSA.1/SD | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_MSA.3/SD | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_SMF.1/SD | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_SMR.1/SD | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FTP_ITC.1/SC | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FCO_NRO.2/SC | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FDP_IFC.2/SC | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FDP_IFF.1/SC | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_MSA.1/SC | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_MSA.3/SC | X | X | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FMT_SMF.1/SC | X | X | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FIA_UID.1/SC | X | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FIA_UAU.1/SC | X | X | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FIA_UAU.4/SC | X | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| FDP_ACC.2/FIREWALL | | | | | | | | | X | | | | X | | | | | | X | | | | | | | | | | | | | | | |
| FDP_ACF.1/FIREWALL | | | | | | | | | X | | | X | X | | | | | | X | | | | | | | | | | | | | | | |
| FDP_IFC.1/JCVM | | | | | | | | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | |
| FDP_IFF.1/JCVM | | | | | | | | | X | X | X | | | | | | | | | | | | | | | | | | | | | | | |
| FDP_RIP.1/OBJECTS | | | | | | | | | | | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FMT_MSA.1/JCRE | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FMT_MSA.1/JCVM | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_MSA.2/FIREWALL_JCVM | | | | | | | | | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_MSA.3/FIREWALL | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_MSA.3/JCVM | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_SMF.1 | | | | | | | | | X | | | | | | X | | | | | | | | | | | | | | | | | | |
| FMT_SMR.1 | | | | | | | | | X | | | | | | X | | | | | | | | | | | | | | | | | | |
| FCS_CKM.1/TDES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.1/AES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.1/RSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.1/ECDSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.1/HMAC | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.2/TDES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.2/AES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.2/RSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.2/ECDSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.2/HMAC | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.3/TDES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.3/AES | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.3/RSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.3/ECDSA | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.3/HMAC | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_CKM.4 | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_COP.1/TDES_CIPHER | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_COP.1/TDES_MAC | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/AES_CIPHER | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_COP.1/AES_MAC | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/RSA_SIGN | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/RSA_CIPHER | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FCS_COP.1/ECDSA_SIGN | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/ECDH | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/Hash | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | |
| FCS_COP.1/HMAC | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | | | | |
| FCS_COP.1/CRC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | |
| FCS_RND.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| FDP_RIP.1/ABORT | | | | | | | | | | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FDP_RIP.1/APDU | | | | | | | | | X | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FDP_RIP.1/bArray | | | | | | | | | X | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FDP_RIP.1/KEYS | | | | | | | | | | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FDP_RIP.1/TRANSIENT | | | | | | | | | | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| FDP_ROL.1/FIREWALL | | | | | | | | | | | | X | | X | | | | X | X | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | | | | | | | | | | X | | X | X | | | | | | | | | | | | | | | | | | |
| FDP_SDI.2 | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| FPR_UNO.1 | | | | | | | | | | | | | | | | X | X | X | | | | | | | | | | | | | | | |
| FPT_FLS.1/JCS | | | | | | | | | | | | X | | X | X | | | | | | | | | | X | | | | | | | | |
| FPT_TDC.1 | | | | | | | | | | | | X | | | | | | | | | | | | | | | | | | | | | |
| FIA_ATD.1/AID | | | | | | | | X | | | | X | | | | | | | | | | | | | | | | | | | | | |
| FIA_UID.2/AID | | | | | | | | X | | | | | | | | | | | | | | | | | | | | | | | | | |
| FIA_USB.1/AID | | | | | | | | X | | | | X | | | | | | | | | | | | | | | | | | | | | |
| FMT_MTD.1/JCRE | | | | | | | | X | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FMT_MTD.3/JCRE | | | | | | | | X | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FDP_ITC.2/Installer | | | | | | | | X | X | | | X | | | | | | | | | | | | | X | | | | | | | | |
| FMT_SMR.1/Installer | | | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FPT_FLS.1/Installer | | | | | | | | | | | | X | | X | X | | | | | | | | | | X | | | | | | | | |
| FPT_RCV.3/Installer | | | | | | | | | | | | X | | X | | | | | | | X | | X | | | | | | | | | | |
| FDP_ACC.2/ADEL | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | |
| FDP_ACF.1/ADEL | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | | |
| FDP_RIP.1/ADEL | | | | | | | | | | | | | X | | | | X | X | X | | X | | | | | | | | | | | | |
| FMT_MSA.1/ADEL | | | | | | | | X | X | | | | | | | | | | | | X | | | | | | | | | | | | |
| FMT_MSA.3/ADEL | | | | | | | | X | X | | | | | | | | | | | | X | | | | | | | | | | | | |
| FMT_SMF.1/ADEL | | | | | | | | X | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FMT_SMR.1/ADEL | | | | | | | | | X | | | | | X | | | | | | | X | | | | | | | | | | | | |
| FPT_FLS.1/ADEL | | | | | | | | | | | | X | | X | X | | | | | | X | | | | | | | | | | | | |
| FDP_RIP.1/ODEL | | | | | | | | | | | | | X | | | | X | X | X | X | | | | | | | | | | | | | |
| FPT_FLS.1/ODEL | | | | | | | | | | | | X | | X | X | | | | | X | | | | | | | | | | | | | |
| FCO_NRO.2/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FDP_IFC.2/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FDP_IFF.1/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FDP_UIT.1/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FIA_UID.1/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FMT_MSA.1/CM | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_MSA.3/CM | | | | | | | | X | X | | | | | | | | | | | | | | | | | | | | | | | | |
| FMT_SMF.1/CM | | | | | | | | X | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FMT_SMR.1/CM | | | | | | | | | X | | | | | X | | | | | | | | | | | | | | | | | | | |
| FTP_ITC.1/CM | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | | | | |
| FPT_RCV.3/OS | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | | | | | |
| FPT_RCV.4/OS | | | | | | | X | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FPT_FLS.1/SecureAPI | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | | | |
| FPT_ITT.1/SecureAPI | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | | | |
| FPR_UNO.1/SecureAPI | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | | | |
| FMT_SMR.1/GemActivate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X | X |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **FMT_SMF.1/GemActivate** | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X | X | |
| **FMT_MOF.1/GemActivate** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X | |
| **FMT_MSA.1/GemActivate** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X | |
| **FMT_MTD.1/GemActivate** | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| **FDP_ACC.1/GemActivate_DAP** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| **FDP_ACF.1/ GemActivate_DAP** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| **FMT_MSA.3/GemActivate_DAP** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| **FIA_ATD.1/AID_Patch** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X |

***Table 5: TOE Security Objectives coverage by Security Functional Requirements – Mapping table***

### *9.3.2. TOE security objectives coverage – Rationale*

**O.CARD-MANAGEMENT**
The security objective O.CARD-MANAGEMENT is met by the following SFRs:

- FDP_UIT.1/CCM enforces the Secure Channel Protocol information flow control policy and the Security Domain access control policy to ensure the integrity of card management operations.
- FDP_ROL.1/CCM ensures that card management operations may be cleanly aborted.
- FDP_ITC.2/CCM enforces the Firewall access control policy and the Secure Channel Protocol information flow policy when importing card management data.
- FPT_FLS.1/CCM preserves a secure state when failures occur.
- All SFRs related to Security Domains (FDP_ACC.1/SD, FDP_ACF.1/SD, FMT_MSA.1/SD, FMT_MSA.3/SD, FMT_SMF.1/SD, FMT_SMR.1/SD) cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
- All SFRs related to the secure channel (FMT_MSA.1/SC, FMT_MSA.3/SC, FMT_SMF.1/SC, FIA_UAU.1/SC, FTP_ITC.1/SC, FCO_NRO.2/SC, FDP_IFC.2/SC, FDP_IFF.1/SC, FIA_UID.1/SC, FIA_UAU.4/SC) support this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

**O.DOMAIN-RIGHTS**
The security objective O.DOMAIN-RIGHTS is met by the following SFRs:

- All SFRs related to Security Domains (FDP_ACC.1/SD, FDP_ACF.1/SD, FMT_MSA.1/SD, FMT_MSA.3/SD, FMT_SMF.1/SD, FMT_SMR.1/SD) cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
- All SFRs related to the secure channel (FMT_MSA.1/SC, FMT_MSA.3/SC, FMT_SMF.1/SC, FIA_UAU.1/SC, FTP_ITC.1/SC, FCO_NRO.2/SC, FDP_IFC.2/SC, FDP_IFF.1/SC, FIA_UID.1/SC, FIA_UAU.4/SC) support this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

**O.APPLI-AUTH**
The security objective O.APPLI-AUTH is met by the following SFRs:

- FDP_ROL.1/CCM ensures that card management operations may be cleanly aborted.
- FPT_FLS.1/CCM preserves a secure state when failures occur.
- FCS_COP.1/DAP ensures that the loaded Executable Application is legitimate by specifying the algorithm to be used in order to verify the DAP signature of the Verification Authority.

**O.COMM_AUTH**
This security objective is covered by the following security functional requirements:

- FTP_ITC.1/SC which ensures the origin of card administration commands.
- FMT_SMR.1/SD specifies the authorized identified roles enabling to send and authenticate card management commands.
- FDP_IFC.2/SC and FDP_IFF.1/SC enforces the Secure Channel Protocol information flow control policy to ensure the origin of administration requests.
- FMT_MSA.1/SC and FMT_MSA.3/SC covers indirectly this security objective by specifying security attributes enabling to authenticate card management requests.
- FIA_UID.1/SC and FIA_UAU.1/SC specify the actions that can be performed before authenticating the origin of the APDU commands that the (U)SIM card receives.
- The security functional requirement FCS_COP.1 defined in [JCRE] supports also this security objective by specifying secure cryptographic algorithm that shall be used to determine the origin of the card management commands.

**O.COMM_INTEGRITY**
This security objective is covered by the following security functional requirements:

- FTP_ITC.1/SC which ensures the integrity of card management commands.

- FMT_SMF.1/SC specifies the actions activating the integrity check on the card management commands.
- FMT_SMR.1/SD defines the roles enabling to send and authenticate the card management requests for which the integrity has to be ensured.
- FDP_IFC.2/SC and FDP_IFF.1/SC enforces the Secure Channel Protocol information flow control policy to guarantee the integrity of administration requests.
- FMT_MSA.1/SC and FMT_MSA.3/SC covers indirectly this security objective by specifying security attributes enabling to guarantee the integrity of card management requests.
- The security functional requirement FCS_COP.1 defined in [JCRE] supports also this security objective by specifying secure cryptographic algorithm that shall be used to ensure the integrity of the card management commands.

**O.COMM_CONFIDENTIALITY**
This security objective is covered by the following security functional requirements:

- FTP_ITC.1/SC which ensures the confidentiality of card management commands.
- FMT_SMF.1/SC specifies the actions ensuring the confidentiality of the card management commands.
- FMT_SMR.1/SD defines the roles enabling to send and authenticate the card management requests for which the confidentiality has to be ensured.
- FDP_IFC.2/SC and FDP_IFF.1/SC enforces the Secure Channel Protocol information flow control policy to guarantee the confidentiality of administration requests.
- FMT_MSA.1/SC and FMT_MSA.3/SC covers indirectly this security objective by specifying security attributes enabling to guarantee the confidentiality of card management requests by decrypting those requests and imposing management conditions on that attributes.
- The security functional requirement FCS_COP.1 defined in [JCRE] supports also this security objective by specifying secure cryptographic algorithm that shall be used to ensure the confidentiality of the card management commands.

**O.SCP-SUPPORT**
The SCP is a part of the TOE supporting TSFs of the upper layer of the TOE, especially for recovery operations as FPT_RCV.4/OS.

**O.SID**
Subjects' identity is AID-based (applets, packages), and is met by the following SFRs: FDP_ITC.2/Installer, FIA_ATD.1/AID, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_MSA.1/ADEL, FMT_MSA.1/CM, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE and FMT_MTD.3/JCRE. Lastly, installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities (FIA_UID.2/AID, FIA_USB.1/AID).

**O.FIREWALL**
This objective is met by the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) and the functional requirement FDP_ITC.2/Installer. The functional requirements of the class FMT (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMR.1/CM, FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM) also indirectly contribute to meet this objective.

**O.GLOBAL_ARRAYS_CONFID**
Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer and the global byte array input parameter (bArray) to an applet's install method. The clearing requirement of these arrays is met by (FDP_RIP.1/APDU and FDP_RIP.1/bArray respectively). The JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.

**O.GLOBAL_ARRAYS_INTEG**
This objective is met by the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM), which prevents an application from keeping a pointer to the APDU buffer of the card or to the global byte array of the applet's install method. Such a pointer could be used to access and modify it when the buffer is being used by another application.

**O.NATIVE**
This security objective is covered by FDP_ACF.1/FIREWALL: the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective OE.APPLET, which uphold the assumption A.APPLET.

**O.OPERATE**
The TOE is protected in various ways against applets' actions (FPT_TDC.1), the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, and is able to detect and block various failures or security violations during usual working (FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FPT_FLS.1/Installer, FAU_ARP.1). Its security-critical parts and procedures are also protected: safe recovery from failure is ensured (FPT_RCV.3/Installer), applets' installation may be cleanly aborted (FDP_ROL.1/FIREWALL), communication with external users and their internal subjects is well-controlled (FDP_ITC.2/Installer, FIA_ATD.1/AID, FIA_USB.1/AID) to prevent alteration of TSF data (also protected by components of the FPT class). Almost every objective and/or functional requirement indirectly contributes to this one too.

Application note: Startup of the TOE (TSF-testing) can be covered by FPT_TST.1. This SFR component is not mandatory in [JCRE301], but appears in most of security requirements documents for masked applications. Testing could also occur randomly. Self-tests may become mandatory in order to comply with FIPS certification [FIPS 140-2].

**O.REALLOCATION**
This security objective is satisfied by the following SFRs: FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ADEL, which imposes that the contents of the re-allocated block shall always be cleared before delivering the block.

**O.RESOURCES**
The TSFs detects stack/memory overflows during execution of applications (FAU_ARP.1, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FPT_FLS.1/Installer). Failed installations are not to create memory leaks (FDP_ROL.1/FIREWALL, FPT_RCV.3/Installer) as well. Memory management is controlled by the TSF (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1, FMT_SMF.1 FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM and FMT_SMR.1/CM).

**O.ALARM**
This security objective is met by FPT_FLS.1/Installer, FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL which guarantee that a secure state is preserved by the TSF when failures occur, and FAU_ARP.1 which defines TSF reaction upon detection of a potential security violation.

**O.CIPHER**
This security objective is directly covered by FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.2/TDES, FCS_CKM.2/AES, FCS_CKM.2/RSA, FCS_CKM.2/ECDSA, FCS_CKM.2/HMAC, FCS_CKM.3/AES, FCS_CKM.3/TDES, FCS_CKM.3/RSA, FCS_CKM.3/ECDSA, FCS_CKM.3/HMAC, FCS_CKM.4, FCS_COP.1/TDES_CIPHER, FCS_COP.1/AES_CIPHER and FCS_COP.1/RSA_CIPHER. The SFR FPR_UNO.1 contributes in covering this security objective and controls the observation of the cryptographic operations which may be used to disclose the keys.

**O.KEY-MNGT**
This relies on the same security functional requirements as O.CIPHER, plus FDP_RIP.1 and FDP_SDI.2 as well. Precisely it is met by the following components: FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.2/TDES,

FCS_CKM.2/AES, FCS_CKM.2/RSA, FCS_CKM.2/ECDSA, FCS_CKM.2/HMAC, FCS_CKM.3/AES, FCS_CKM.3/TDES, FCS_CKM.3/RSA, FCS_CKM.3/ECDSA, FCS_CKM.3/HMAC, FCS_CKM.4, FCS_COP.1/TDES_MAC, FCS_COP.1/AES_MAC, FCS_COP.1/RSA_SIGN, FCS_COP.1/ECDSA_SIGN, FCS_COP.1/ECDH, FCS_COP.1/RSA_CIPHER, FCS_COP.1/HMAC, FCS_COP.1/TDES_CIPHER, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL and FDP_RIP.1/TRANSIENT.

**O.PIN-MNGT**
This security objective is ensured by FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FPR_UNO.1, FDP_ROL.1/FIREWALL and FDP_SDI.2 security functional requirements. The TSFs behind these are implemented by API classes. The firewall security functions FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL shall protect the access to private and internal data of the objects.

**O.TRANSACTION**
Directly met by FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT and FDP_RIP.1/OBJECTS (more precisely, by the element FDP_RIP.1.1/ABORT).

**O.OBJ-DELETION**
This security objective specifies that deletion of objects is secure. The security objective is met by the security functional requirements FDP_RIP.1/ODEL and FPT_FLS.1/ODEL.

**O.DELETION**
This security objective specifies that applet and package deletion must be secure. The non-introduction of security holes is ensured by the ADEL access control policy (FDP_ACC.2/ADEL, FDP_ACF.1/ADEL). The integrity and confidentiality of data that does not belong to the deleted applet or package is a by-product of this policy as well. Non-accessibility of deleted data is met by FDP_RIP.1/ADEL and the TSFs are protected against possible failures of the deletion procedures (FPT_FLS.1/ADEL, FPT_RCV.3/Installer). The security functional requirements of the class FMT (FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL) included in the group ADELG also contribute to meet this objective.

**O.LOAD**
This security objective specifies that the loading of a package into the card must be secure. Evidence of the origin of the package is enforced (FCO_NRO.2/CM) and the integrity of the corresponding data is under the control of the PACKAGE LOADING information flow policy (FDP_IFC.2/CM, FDP_IFF.1/CM) and FDP_UIT.1/CM. Appropriate identification (FIA_UID.1/CM) and transmission mechanisms are also enforced (FTP_ITC.1/CM).

**O.INSTALL**
This security objective specifies that installation of applets must be secure. Security attributes of installed data are under the control of the FIREWALL access control policy (FDP_ITC.2/Installer), and the TSFs are protected against possible failures of the installer (FPT_FLS.1/Installer, FPT_RCV.3/Installer).

**O.SCP.IC** The SCP.IC is a part of the TOE supporting TSFs of the upper layer of the TOE and more specially FPT_FLS.1/JCS

**O.SCP.RECOVERY**
The SCP is a part of the TOE supporting TSFs of the upper layer of the TOE, especially for recovery operations as FPT_RCV.3/OS.

**O.Secure_API**
The security objective is met by the following SFR FPT_FLS.1/SecureAPI, FPT_ITT.1/SecureAPI and FPR_UNO.1/SecureAPI.

**O.RND**
The security objective O.RND is met by the following SFR FCS_RND.1.

**O.JCAPI-Services**
The security objective is met by the following SFR FCS_COP.1/Hash and FCS_COP.1/CRC.

**O.REMOTE_SERVICE_AUDIT**
The security objective is met by the following SFR: FMT_MTD.1/GemActivate, FMT_SMR.1/GemActivate and FMT_SMF.1/GemActivate.

**O.REMOTE_SERVICE_ACTIVATION**
The security objective is met by the following SFR: FMT_SMR.1/GemActivate, FMT_SMF.1/GemActivate, FMT_MOF.1/GemActivate and FMT_MSA.1/GemActivate.

**O.Secure_Load_ACode**
The security objective is met by the following SFRs:

- FDP_UIT.1/CCM, FDP_ROL.1/CCM, FDP_ITC.2/CCM, FPT_FLS.1/CCM as defined for O.CARD-MANAGEMENT but applied to patch.
- All SFRs related to Security Domains (FDP_ACC.1/SD, FDP_ACF.1/SD, FMT_MSA.1/SD, FMT_MSA.3/SD, FMT_SMF.1/SD, FMT_SMR.1/SD) as defined for O.CARD-MANAGEMENT but applied to patch.
- All SFRs related to the secure channel (FMT_MSA.1/SC, FMT_MSA.3/SC, FMT_SMF.1/SC, FIA_UAU.1/SC, FTP_ITC.1/SC, FCO_NRO.2/SC, FDP_IFC.2/SC, FDP_IFF.1/SC, FIA_UID.1/SC, FIA_UAU.4/SC) as defined for O.CARD-MANAGEMENT but applied to patch.
- FMT_SMR.1/GemActivate, FMT_SMF.1/GemActivate,FMT_MSA.1/GemActivate, FMT_MOF.1/GemActivate as defined for O.REMOTE_SERVICE_ACTIVATION but applied to patch.
- FCS_COP.1/DAP, FDP_ACC.1/GemActivate_DAP, FDP_ACF.1/ GemActivate_DAP and FMT_MSA.3/GemActivate_DAP to ensure authenticity and integrity of patch loading through the DAP mechanism.

**O.Secure_AC_Activation**
The security objective is met by the following SFRs:

- FDP_UIT.1/CCM, FDP_ROL.1/CCM, FDP_ITC.2/CCM, FPT_FLS.1/CCM as defined for O.CARD-MANAGEMENT but applied to patch.
- FMT_SMR.1/GemActivate, FMT_SMF.1/GemActivate,FMT_MSA.1/GemActivate, FMT_MOF.1/GemActivate as defined for O.REMOTE_SERVICE_ACTIVATION but applied to patch.

**O.TOE_Identification**
The security objective is met by the following SFRs: FDP_ROL.1/CCM, FDP_ITC.2/CCM as defined for O.CARD-MANAGEMENT but applied to patch, and FIA_ATD.1/AID_Patch.

### *9.3.3. SFR dependency rationale*

| Security Functional Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **FDP_UIT.1/CCM** | (FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1) | FDP_ACC.1/SD<br>FTP_ITC.1/SC |
| **FDP_ROL.1/CCM** | (FDP_ACC.1 or FDP_IFC.1) | FDP_ACC.1/SD |
| **FDP_ITC.2/CCM** | (FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1) | FDP_ACC.1/SD<br>FTP_ITC.1/SC<br>**See rationale** |
| **FPT_FLS.1/CCM** | No dependencies | |

| Security Functional Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **FCS_COP.1/DAP** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FDP_ITC.2/CCM<br>FCS_CKM.4 |
| **FDP_ACC.1/SD** | (FDP_ACF.1) | FDP_ACF.1/SD |
| **FDP_ACF.1/SD** | (FDP_ACC.1) and (FMT_MSA.3) | FDP_ACC.1/SD<br>FMT_MSA.3/SD |
| **FMT_MSA.1/SD** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_ACC.1/SD<br>FMT_SMF.1/SD<br>FMT_SMR.1/SD |
| **FMT_MSA.3/SD** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/SD<br>FMT_SMR.1/SD |
| **FMT_SMF.1/SD** | No dependencies | |
| **FMT_SMR.1/SD** | (FIA_UID.1) | FIA_UID.1/SC |
| **FTP_ITC.1/SC** | No dependencies | |
| **FCO_NRO.2/SC** | (FIA_UID.1) | FIA_UID.1/SC |
| **FDP_IFC.2/SC** | (FDP_IFF.1) | FDP_IFF.1/SC |
| **FDP_IFF.1/SC** | (FDP_IFC.1) and (FMT_MSA.3) | FDP_IFC.2/SC<br>FMT_MSA.3/SC |
| **FMT_MSA.1/SC** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_ACC.1/SD<br>FMT_SMR.1/SD<br>FMT_SMF.1/SC |
| **FMT_MSA.3/SC** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_SMR.1/SD<br>FMT_MSA.1/SC |
| **FMT_SMF.1/SC** | No dependencies | |
| **FIA_UID.1/SC** | No dependencies | |
| **FIA_UAU.1/SC** | (FIA_UID.1) | FIA_UID.1/SC |
| **FIA_UAU.4/SC** | No dependencies | |
| **FDP_ACC.2/FIREWALL** | (FDP_ACF.1) | FDP_ACF.1/FIREWALL |
| **FDP_ACF.1/FIREWALL** | (FDP_ACC.1) and (FMT_MSA.3) | FDP_ACC.2/FIREWALL<br>FMT_MSA.3/FIREWALL |
| **FDP_IFC.1/JCVM** | (FDP_IFF.1) | FDP_IFF.1/JCVM |
| **FDP_IFF.1/JCVM** | (FDP_IFC.1) and (FMT_MSA.3) | FDP_IFC.1/JCVM,<br>FMT_MSA.3/JCVM |
| **FDP_RIP.1/OBJECTS** | No dependencies | |
| **FMT_MSA.1/JCRE** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_ACC.2/FIREWALL,<br>FMT_SMR.1<br>**See rationale** |
| **FMT_MSA.1/JCVM** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_ACC.2/FIREWALL<br>FDP_IFC.1/JCVM<br>FMT_SMF.1<br>FMT_SMR.1 |
| **FMT_MSA.2/FIREWALL_JCVM** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.1) and (FMT_SMR.1) | FDP_ACC.2/FIREWALL<br>FDP_IFC.1/JCVM<br>FMT_MSA.1/JCRE<br>FMT_MSA.1/JCVM<br>FMT_SMR.1 |
| **FMT_MSA.3/FIREWALL** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/JCRE<br>FMT_MSA.1/JCVM<br>FMT_SMR.1 |
| **FMT_MSA.3/JCVM** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/JCVM<br>FMT_SMR.1 |
| **FMT_SMF.1** | No dependencies | |
| **FMT_SMR.1** | (FIA_UID.1) | FIA_UID.2/AID |
| **FCS_CKM.1/TDES** | (FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4) | FCS_CKM.2/TDES<br>FCS_COP.1/TDES_CIPHER<br>FCS_COP.1/TDES_MAC<br>FCS_CKM.4 |
| **FCS_CKM.1/AES** | (FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4) | FCS_CKM.2/AES<br>FCS_COP.1/AES_CIPHER<br>FCS_COP.1/AES_MAC<br>FCS_CKM.4 |
| **FCS_CKM.1/RSA** | (FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4) | FCS_CKM.2/RSA<br>FCS_COP.1/RSA_SIGN<br>FCS_COP.1/RSA_CIPHER |

| Security Functional Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| | | FCS_CKM.4 |
| **FCS_CKM.1/ECDSA** | (FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4) | FCS_CKM.2/ECDSA<br>FCS_COP.1/ECDSA_SIGN<br>FCS_COP.1/ECDH<br>FCS_CKM.4 |
| **FCS_CKM.1/HMAC** | (FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4) | FCS_CKM.2/HMAC<br>FCS_COP.1/HMAC<br>FCS_CKM.4 |
| **FCS_CKM.2/TDES** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/TDES<br>FCS_CKM.4 |
| **FCS_CKM.2/AES** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/AES<br>FCS_CKM.4 |
| **FCS_CKM.2/RSA** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/RSA<br>FCS_CKM.4 |
| **FCS_CKM.2/ECDSA** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/ECDSA<br>FCS_CKM.4 |
| **FCS_CKM.2/HMAC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/HMAC<br>FCS_CKM.4 |
| **FCS_CKM.3/TDES** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/TDES<br>FCS_CKM.4 |
| **FCS_CKM.3/AES** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/AES<br>FCS_CKM.4 |
| **FCS_CKM.3/RSA** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/RSA<br>FCS_CKM.4 |
| **FCS_CKM.3/ECDSA** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/ECDSA<br>FCS_CKM.4 |
| **FCS_CKM.3/HMAC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/HMAC<br>FCS_CKM.4 |
| **FCS_CKM.4** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) | FCS_CKM.1/TDES<br>FCS_CKM.1/AES<br>FCS_CKM.1/RSA<br>FCS_CKM.1/ECDSA<br>FCS_CKM.1/HMAC |
| **FCS_COP.1/TDES_CIPHER** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/TDES<br>FCS_CKM.4 |
| **FCS_COP.1/TDES_MAC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/TDES<br>FCS_CKM.4 |
| **FCS_COP.1/AES_CIPHER** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/AES<br>FCS_CKM.4 |
| **FCS_COP.1/AES_MAC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/AES<br>FCS_CKM.4 |
| **FCS_COP.1/RSA_SIGN** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/RSA<br>FCS_CKM.4 |
| **FCS_COP.1/RSA_CIPHER** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/RSA<br>FCS_CKM.4 |
| **FCS_COP.1/ECDSA_SIGN** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/ECDSA<br>FCS_CKM.4 |
| **FCS_COP.1/ECDH** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/ECDSA<br>FCS_CKM.4 |
| **FCS_COP.1/HMAC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | FCS_CKM.1/HMAC<br>FCS_CKM.4 |
| **FDP_RIP.1/ABORT** | No dependencies | |
| **FDP_RIP.1/APDU** | No dependencies | |
| **FDP_RIP.1/bArray** | No dependencies | |
| **FDP_RIP.1/KEYS** | No dependencies | |
| **FDP_RIP.1/TRANSIENT** | No dependencies | |
| **FDP_ROL.1/FIREWALL** | (FDP_ACC.1 or FDP_IFC.1) | FDP_ACC.2/FIREWALL<br>FDP_IFC.1/JCVM |
| **FAU_ARP.1** | (FAU_SAA.1) | **See rationale** |
| **FDP_SDI.2** | No dependencies | |
| **FPR_UNO.1** | No dependencies | |
| **FPT_FLS.1/JCS** | No dependencies | |
| **FPT_TDC.1** | No dependencies | |

| Security Functional Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **FIA_ATD.1/AID** | No dependencies | |
| **FIA_UID.2/AID** | No dependencies | |
| **FIA_USB.1/AID** | (FIA_ATD.1) | FIA_ATD.1/AID |
| **FMT_MTD.1/JCRE** | (FMT_SMF.1) and (FMT_SMR.1) | FMT_SMF.1<br>FMT_SMR.1 |
| **FMT_MTD.3/JCRE** | (FMT_MTD.1) | FMT_MTD.1/JCRE |
| **FDP_ITC.2/Installer** | (FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1) | FDP_IFC.2/CM<br>FTP_ITC.1/CM<br>FPT_TDC.1 |
| **FMT_SMR.1/Installer** | (FIA_UID.1) | **See rationale** |
| **FPT_FLS.1/Installer** | No dependencies | |
| **FPT_RCV.3/Installer** | (AGD_OPE.1) | AGD_OPE.1 |
| **FDP_ACC.2/ADEL** | (FDP_ACF.1) | FDP_ACF.1/ADEL |
| **FDP_ACF.1/ADEL** | (FDP_ACC.1) and (FMT_MSA.3) | FDP_ACC.2/ADEL<br>FMT_MSA.3/ADEL |
| **FDP_RIP.1/ADEL** | No dependencies | |
| **FMT_MSA.1/ADEL** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_ACC.2/ADEL<br>FMT_SMF.1/ADEL<br>FMT_SMR.1/ADEL |
| **FMT_MSA.3/ADEL** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/ADEL<br>FMT_SMR.1/ADEL |
| **FMT_SMF.1/ADEL** | No dependencies | |
| **FMT_SMR.1/ADEL** | (FIA_UID.1) | **See rationale** |
| **FPT_FLS.1/ADEL** | No dependencies | |
| **FDP_RIP.1/ODEL** | No dependencies | |
| **FPT_FLS.1/ODEL** | No dependencies | |
| **FCO_NRO.2/CM** | (FIA_UID.1) | FIA_UID.1/CM |
| **FDP_IFC.2/CM** | (FDP_IFF.1) | FDP_IFF.1/CM |
| **FDP_IFF.1/CM** | (FDP_IFC.1) and (FMT_MSA.3) | FDP_IFC.2/CM<br>FMT_MSA.3/CM |
| **FDP_UIT.1/CM** | (FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1) | FDP_IFC.2/CM<br>FTP_ITC.1/CM |
| **FIA_UID.1/CM** | No dependencies | |
| **FMT_MSA.1/CM** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FDP_IFC.2/CM<br>FMT_SMF.1/CM<br>FMT_SMR.1/CM |
| **FMT_MSA.3/CM** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/CM<br>FMT_SMR.1/CM |
| **FMT_SMF.1/CM** | No dependencies | |
| **FMT_SMR.1/CM** | (FIA_UID.1) | FIA_UID.1/CM |
| **FTP_ITC.1/CM** | No dependencies | |
| **FPT_RCV.3/OS** | (AGD_OPE.1) | AGD_OPE.1 |
| **FPT_RCV.4/OS** | No dependencies | |
| **FCS_COP.1/Hash** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | **See rationale** |
| **FCS_COP.1/CRC** | (FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4) | **See rationale** |
| **FCS_RND.1** | No dependencies | |
| **FPT_FLS.1/SecureAPI** | No dependencies | |
| **FPT_ITT.1/SecureAPI** | No dependencies | |
| **FPR_UNO.1/SecureAPI** | No dependencies | |
| **FMT_SMR.1/GemActivate** | (FIA_UID.1) | FIA_UID.1/CM<br>FIA_UID.2/AID<br>FIA_UID.1/SC |
| **FMT_SMF.1/GemActivate** | No dependencies | |
| **FMT_MOF.1/GemActivate** | (FMT_SMF.1) and (FMT_SMR.1) | FMT_SMR.1/GemActivate<br>FMT_SMF.1/GemActivate |
| **FMT_MSA.1/GemActivate** | (FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1) | FMT_SMR.1/GemActivate<br>FMT_SMF.1/GemActivate<br>**See rationale** |
| **FMT_MTD.1/GemActivate** | (FMT_SMF.1) and (FMT_SMR.1) | FMT_SMR.1/GemActivate,<br>FMT_SMF.1/GemActivate |

| Security Functional Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **FDP_ACC.1/GemActivate_DAP** | (FDP_ACF.1) | FDP_ACF.1/ GemActivate_DAP |
| **FDP_ACF.1/GemActivate_DAP** | (FDP_ACC.1) and (FMT_MSA.3) | FDP_ACC.1/GemActivate_DAP<br>FMT_MSA.3/GemActivate_DAP |
| **FMT_MSA.3/GemActivate_DAP** | (FMT_MSA.1) and (FMT_SMR.1) | FMT_MSA.1/GemActivate<br>FMT_SMR.1/GemActivate |
| **FIA_ATD.1/AID_Patch** | No dependencies | |

**Rationale for the exclusion of dependencies:**

- **The dependency FPT_TDC.1 of FDP_ITC.2/CCM is unsupported.**
  See rationale in PP.
- **The dependency FMT_SMF.1 of FMT_MSA.1/JCRE is unsupported.**
  The dependency between FMT_MSA.1/JCRE and FMT_SMF.1 is not satisfied because no management functions are required for the Java Card RE.
- **The dependency FAU_SAA.1 of FAU_ARP.1 is unsupported**
  The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a "potential security violation" generates an audit event. On the contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in this ST.
- **The dependency FIA_UID.1 of FMT_SMR.1/Installer is unsupported**
  This ST does not require the identification of the "installer" since it can be considered as part of the TSF.
- **The dependency FIA_UID.1 of FMT_SMR.1/ADEL is unsupported**
  This ST does not require the identification of the "deletion manager" since it can be considered as part of the TSF.
- **The dependencies of FCS_COP.1/Hash are unsupported**
  Hash operation does not require any key.
- **The dependencies of FCS_COP.1/CRC are unsupported**
  CRC operations do not require any key.
- **The dependency FDP_ACC.1 or FDP_IFC.1 of FMT_MSA.1/GemActivate is unsupported**
  GemActivate Access Control policy is dedicated to TOE services linked to TSF data, therefore no user data is used requiring link to FDP family.

### *9.3.4. SAR – Evaluation Assurance Level Rationale*

The EAL4 package and addition of ALC_DVS.2 and AVA_VAN.5 are required by [PP-USIM].

### *9.3.5. SAR – Dependency rationale*

| Security Assurance Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **ADV_ARC.1** | (ADV_FSP.1) and (ADV_TDS.1) | ADV_FSP.4<br>ADV_TDS.3 |
| **ADV_FSP.4** | (ADV_TDS.1) | ADV_TDS.3 |
| **ADV_TDS.3** | (ADV_FSP.4) | ADV_FSP.4 |
| **ADV_IMP.1** | (ADV_TDS.3) and (ALC_TAT.1) | ADV_TDS.3<br>ALC_TAT.1 |
| **AGD_OPE.1** | (ADV_FSP.1) | ADV_FSP.4 |
| **AGD_PRE.1** | No dependencies | |

| Security Assurance Requirement | CC dependencies | Satisfied dependencies |
|---|---|---|
| **ALC_CMC.4** | (ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1) | ALC_CMS.4<br>ALC_DVS.2<br>ALC_LCD.1 |
| **ALC_CMS.4** | No dependencies | |
| **ALC_DEL.1** | No dependencies | |
| **ALC_DVS.2** | No dependencies | |
| **ALC_LCD.1** | No dependencies | |
| **ALC_TAT.1** | (ADV_IMP.1) | ADV_IMP.1 |
| **ASE_CCL.1** | (ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1) | ASE_ECD.1<br>ASE_INT.1<br>ASE_REQ.2 |
| **ASE_ECD.1** | No dependencies | |
| **ASE_INT.1** | No dependencies | |
| **ASE_OBJ.2** | (ASE_SPD.1) | ASE_SPD.1 |
| **ASE_REQ.2** | (ASE_ECD.1) and (ASE_OBJ.2) | ASE_ECD.1<br>ASE_OBJ.2 |
| **ASE_SPD.1** | No dependencies | |
| **ASE_TSS.1** | (ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1) | ADV_FSP.4<br>ASE_INT.1<br>ASE_REQ.2 |
| **ATE_COV.2** | (ADV_FSP.2) and (ATE_FUN.1) | ADV_FSP.4<br>ATE_FUN.1 |
| **ATE_DPT.1** | (ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1) | ADV_ARC.1<br>ADV_TDS.3<br>ATE_FUN.1 |
| **ATE_FUN.1** | (ATE_COV.1) | ATE_COV.2 |
| **ATE_IND.2** | (ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1) | ADV_FSP.4<br>AGD_OPE.1<br>AGD_PRE.1<br>ATE_COV.2<br>ATE_FUN.1 |
| **AVA_VAN.5** | (ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1) | ADV_ARC.1<br>ADV_FSP.4<br>ADV_IMP.1<br>ADV_TDS.3<br>AGD_OPE.1<br>AGD_PRE.1<br>ATE_DPT.1 |

The table here-above shows that all SAR dependencies are met.

## 9.4. COMPOSITION TASKS – SFR PART

The following table (see next page) lists the SFRs that are declared in the security target [ST_ST33G1M2], and separates them in relevant platform[1]-SFRs (RP_SFR) and irrelevant platform-SFRs (IP_SFR), as requested in [CCDB]. The table also provides the link between the relevant platform-SFRs and the composite product SFRs.

Note: The additional SFRs defined in [ST_ST33G1M2] and related to DESFire are not included in the present statement of compatibility. Indeed, the present composite evaluation does not target DESFire security.

[1] Using the composition tasks terminology, the platform is the ST33G1M2 chip.

| Platform-SFR | Platform-SFR content | Platform-SFR additional information | RP_SFR | IP_SFR | Composite product SFRs |
|---|---|---|---|---|---|
| FRU_FLT.2 | **Limited fault tolerance**: The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1). | None | X | | FPT_RCV.3/OS<br>FPT_RCV.4/OS |
| FPT_FLS.1 | **Failure with preservation of secure state**: The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur. | None | X | | FPT_RCV.3/OS<br>FPT_RCV.4/OS |
| FMT_LIM.1 (TEST) | The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Limited capability and availability Policy (TEST). | **Limited capability and availability Policy (TEST)**<br>Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks. | X | | FAU_ARP.1<br>FPT_FLS.1/JCS |
| FMT_LIM.2 (TEST) | The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Limited capability and availability Policy (TEST). | | X | | FAU_ARP.1<br>FPT_FLS.1/JCS |
| FAU_SAS.1 | The TSF shall provide the test process before TOE Delivery with the capability to store the Initialization Data and/or Pre-personalization Data and/or supplements of the Security IC Embedded Software in the NVM. | None | X | | No link to TOE SFRs but used for the composite-product identification. |
| FPT_PHP.3 | The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced. | None | X | | FAU_ARP.1<br>FPT_FLS.1/JCS<br>FPT_RCV.3/OS<br>FPT_RCV.4/OS |
| FDP_IFC.1 | The TSF shall enforce the Data Processing Policy on all confidential data when they are processed or transferred by the TSF or by the Security IC Embedded Software. | **Data Processing Policy**<br>User Data and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate | X | | FPR_UNO.1 |
| FDP_ITT.1 | The TSF shall enforce the Data Processing Policy to prevent the disclosure of user data when it is transmitted between | | X | | FPR_UNO.1 |

| Platform-SFR | Platform-SFR content | Platform-SFR additional information | RP_SFR | IP_SFR | Composite product SFRs |
|---|---|---|---|---|---|
| | physically-separated parts of the TOE. | the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software. | | | |
| FPT_ITT.1 | The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.<br>The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE. | | **X** | | FPR_UNO.1 |
| FCS_RNG.1 | The TSF shall provide a physical random number generator that implements: * A total failure test […]<br>The TSF shall provide octets of bits that meet […] | None | **X** | | FCS_RND.1 |
| FCS_COP.1 / EDES | The TSF shall perform: encryption/decryption in CBC and ECB modes, and MAC computation in CBC-MAC, in accordance with a specified cryptographic algorithm: Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES), with cryptographic key sizes of: 56 bits and 168 bits, that meet the following standards: FIPS PUB 46-3, ISO/IEC 9797-1 and ISO/IEC 10116. | None | **X** | | FCS_COP.1/TDES_CIPHER<br>FCS_COP.1/TDES_MAC |
| FCS_COP.1 / AES | The TSF shall perform encryption/decryption in CBC and ECB modes, and CMAC MAC computation, in accordance with a specified cryptographic algorithm: Advanced Encryption Standard and cryptographic key sizes of: 128,192 and 256 bits that meet the following standards: ISO/IEC 10116, NIST SP 800-38B and FIPS PUB 197. | None | **X** | | FCS_COP.1/AES_CIPHER<br>FCS_COP.1/AES_MAC |
| FDP_ACC.2 (MEMORIES) | The TSF shall enforce the Dynamic Memory Access Control Policy on all subjects (software), all objects (data including code stored in memories) and all the operations among subjects and objects covered by the SFP. | **Dynamic Memory Access Control Policy**<br>The TSF must control read, write, execute accesses of software to data, based on the software mode and on the current set of access rights. | **X** | | FDP_ACC.2/FIREWALL |
| FDP_ACF.1 (MEMORIES) | The TSF shall enforce the Dynamic Memory Access Control Policy to objects based on the following: software mode, the object location, the operation to be performed, and the current set of access rights.<br>The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: the operation is allowed if and only if the software | | **X** | | FDP_ACF.1/FIREWALL |

| Platform-SFR | Platform-SFR content | Platform-SFR additional information | RP_SFR | IP_SFR | Composite product SFRs |
|---|---|---|---|---|---|
| | mode, the object location and the operation matches an entry in the current set of access rights.<br>The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.<br>The TSF shall explicitly deny access of subjects to objects based on the following additional rules: in Admin or User configuration, any access (read, write, execute) to the OST ROM is denied, and in User configuration, any write access to the ST NVM is denied. | | | | |
| FMT_MSA.1 (MEMORIES) | The TSF shall enforce the Dynamic Memory Access Control Policy to restrict the ability to modify the security attributes current set of access rights to software running in privileged mode. | | X | | FMT_MSA.1/JCRE<br>FMT_MSA.1/JCVM |
| FMT_MSA.3 (MEMORIES) | The TSF shall enforce the Dynamic Memory Access Control Policy to provide minimally protective default values for security attributes that are used to enforce the SFP.<br>The TSF shall allow none to specify alternative initial values to override the default values when an object or information is created. | | X | | FMT_MSA.3/FIREWALL |
| FMT_SMF.1 (MEMORIES) | The TSF will be able to perform the following management functions: modification of the current set of access rights security attributes by software running in privileged mode, supporting the Dynamic Memory Access Control Policy. | | X | | FMT_SMF.1 |
| FMT_LIM.1 (ADMIN) | The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Limited capability and availability Policy (Admin). | **Limited capability and availability Policy [Admin]**<br>Deploying Loading or Final Test Artifacts after TOE Delivery to final user (phase 7 / USER configuration) does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, stored software to be reconstructed or altered, and no substantial information about construction of TSF to be gathered which may enable other attacks. | X | | FAU_ARP.1<br>FPT_FLS.1/JCS |
| FMT_LIM.2 (ADMIN) | The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Limited capability and availability Policy [Admin]. | | X | | FAU_ARP.1<br>FPT_FLS.1/JCS |
| FDP_ACC.1 (LOADER) | The TSF shall enforce the Loading Access Control Policy on the execution of the Standard Loader instructions and/or the Advanced Loader instructions. | **Loading Access Control Policy** | X | | No direct link to the composite product SFRs, since the IC |

| Platform-SFR | Platform-SFR content | Platform-SFR additional information | RP_SFR | IP_SFR | Composite product SFRs |
|---|---|---|---|---|---|
| FDP_ACF.1 (LOADER) | The TSF shall enforce the Loading Access Control Policy to objects based on the following: an external process may execute the Standard Loader instructions and/or the Advanced Loader instructions, depending on the presentation of valid passwords.<br>The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: the Standard Loader instructions and/or Advanced Loader instructions can be executed only if valid passwords have been presented.<br>The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.<br>The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none. | According to a password control, the TSF grants execution of the instructions of the Standard Loader, Advanced Loader or Profiler. | X | | Loader is no more available after phase 6. However, these six IC TOE SFRs are essential to protect the composite TOE during phases 4, 5 and 6 (covered by the ALC assurance classes) |
| FMT_MSA.1 (LOADER) | The TSF shall enforce the Loading Access Control Policy to restrict the ability to modify the security attributes password to the Standard Loader. | | X | | |
| FMT_MSA.3 (LOADER) | The TSF shall enforce the Loading Access Control Policy to provide restrictive default values for security attributes that are used to enforce the SFP. | | X | | |
| FMT_SMF.1 (LOADER) | The TSF will be able to perform the following management functions: modification of the Standard Loader behavior, by the Advanced Loader, under the Loading Access Control Policy. | | X | | |
| FDP_ITC.1 (LOADER) | The TSF shall enforce the Loading Access Control Policy when importing user data, controlled under the SFP, from outside of the TOE.<br>The TSF shall ignore any security attributes associated with the User data when imported from outside of the TOE.<br>The TSF shall enforce the following rules when importing user data controlled under the SFP from outside of the TOE:<br>- the integrity of the loaded user data is checked at the end of each loading session,<br>- the loaded user data is received encrypted, internally decrypted, then stored into the NVM. | | X | | |

# 10. TOE summary specification

## 10.1. UPTEQ NFC3.2.2_GENERIC V1.0 PLATFORM

**GP.CardContentManagement**
This security function provides the capability and a dedicated flow control for the loading, installation, extradition, registry update, selection and removal of card content and especially executable files and application instances. Such features are offered to the Card Issuer and its business partners, allowing the Card Issuer to delegate card content management to an Application Provider according to privileges assigned to the various security domains on the card. It supports delegated management and it can use DAP or Mandated DAP verification and generation of Reception token. It also checks that only the card management commands specified and allowed at each state of the smart card's life cycle are accepted, and ill-formed ones are rejected with an appropriate error response.

**GP.SecurityDomain**
This security function provides security domain management, as SD creation, SD selection, SD privileges setting and SD deletion in SD hierarchy. It provides means to associate or extradite an application to a security domain in order to provide services (as secure channel) to the dedicated application without sharing the related keys stored in SD. It also provides Keyset Management in SD, with Key Set creation, Key set deletion, key importation, replacement, or deletion in Key Set.
Security Domains are privileged Applications as defined in [GP_22] § 7, holding cryptographic keys to be used to support Secure Channel Protocol operations and/or to authorize card content management functions. There are different types of security domain with dedicated privileges and associated operations: ISD Security domain, Supplementary Security domains, and Controlling Authority Security domains.
ISD Security domain as defined in [GP_22] §7.1.1, is the mandatory Security Domain, implicitly selected if the Application implicitly selectable on the same logical channel of the same card I/O interface is removed. It inherits of the Final Application privilege if the Application with that privilege is removed.
Supplementary Security Domains are privileged Applications with dedicated privileges:

- Token Verification Privilege as described in [GP_22] §9.1.3.1
- Delegated Management Privilege as described in [GP_22] §9.1.3.3
- Global Delete Privilege as described in [GP_22] §9.1.3.4
- Global Lock Privilege as described in [GP_22] §9.1.3.5
- Receipt Generation Privilege as described in [GP_22] §9.1.3.6

Controlling Authority Security Domain is a supplementary Security Domain dedicated to the Controlling Authority with dedicated privileges. It contains Security Domains cryptographic keys needed to confidentially personalize an initial set of Secure Channel Keys of an APSD.

**GP.ISD**
This security function manages the Issuer Security Domain with associated functions and dedicated privileges as defined in [GP_22] §7.1.1.

**GP.CASD**
This security function manages supplementary Controlling Authority Security Domain with associated functions to confidential Card Content Management as defined in [GP22_AmdA].

**GP.VASD**
This security function manages supplementary Verification Authority Security Domain with associated functions to mandated DAP as defined in [GP_22] §9.2.

**GP.SSD**
This security function manages supplementary Security Domains with associated functions and dedicated privileges as defined in [GP_22] §9.1.

**GP.SCP**
This security function manages Secure Channel protocol according to [GP_22] annex E, [GP22_AmdA], [GP22_AmdD] and [TS 102.225].

**GP.SecureChannel**
This security function provides a secure communication channel between a card and an off-card entity during an Application Session according to [GP], [TS 102.225] and [TS 102.226]. It provides an APDU flow control using the Command security level check according to Card Life cycle and type of APDU.
A Secure Channel Session is divided into three sequential phases:

- Secure Channel Initiation when the on-card Application and the off-card entity have exchanged sufficient information enabling them to perform the required cryptographic functions. The Secure Channel Session initiation always includes (at least) the authentication of the off-card entity by the on-card Application; performing also the setting of the Command security level used for the session.
- Secure Channel Operation when the on-card Application and the off-card entity exchange data within the cryptographic protection of the Secure Channel Session. The Secure Channel services offered may vary from one Secure Channel Protocol to the other;
- Secure Channel Termination when either the on-card Application or the off-card entity determines that no further communication is required or allowed via an established Secure Channel Session.

The following services are provided by the Secure Channel as defined in [GP_22] §4.3.2 and §10 using SCP02 or SCP03 or SCP80 or SCP81:

- Entity authentication in which the card or the off-card entity proves its authenticity to the other entity through a cryptographic exchange, based on session key generation and a dedicated flow control; For SCP80, envelope APDU shall contain secured packet structure defined in [TS 102.225] §5 and Anti-replay mechanism is proposed optionally using a counter defined in [TS 102.225] §5.1.4;
- Integrity and authentication in which the receiving entity (the card or off-card entity) ensures that the data being received from the sending entity (respectively the off-card entity or card) actually came from an authenticated entity in the correct sequence and has not been altered;
- Confidentiality in which data being transmitted from the sending entity (the off-card entity or card) to the receiving entity (respectively the card or off-card entity) is not viewable by an unauthenticated entity.

**GP.GPRegistry**
This security function provides access to the GlobalPlatform Registry used for:

- Store card management information;
- Store relevant application management information (e.g., AID, associated Security Domain and Privileges);
- Support card resource management data;
- Store Application Life Cycle information;
- Store card Life Cycle information;
- Track any counters associated with logs.

The content of the GlobalPlatform Registry may be accessed by administrative commands or by applet using a dedicated GP_API.

**GP.PATCH_IDENTIFICATION**
This security function manages patch identification by updating a TAG containing a patch identifier. It is associated to Platform identifier to provide the complete TOE identification.

**JCS.APDUBuffer**
The security function maintains a byte array buffer accessible from any applet context. This buffer is used to transfer incoming APDU header and data bytes as well as outgoing data according to [JCAPI301]. The APDU class API is designed to be transport protocol independent T=0, T=1, T=CL (as defined in ISO 7816-3).
Application note: ADPU buffer is a JCRE temporary entry point object where no associated reference can be stored in a variable or an array component.

**JCS.ByteCodeExecution**
This security function handles applet bytecode execution according to the rules defined in [JCVM301]. The JCVM execution may be summarized in JCVM interpreter start-up, bytecode execution and JCVM interpreter loop. The applet bytecode execution consists in:

- fetching the next bytecode to execute according to the applet's code flow control,
- decoding the next bytecode,
- executing the fetched bytecode.

The JCVM manages 5 types of objects: persistent objects, transient objects, persistent arrays (boolean, byte, short, int or reference), transient arrays (boolean, byte, short, int or reference) and static field images. For each type of object, different types of control are performed.

**JCS.Firewall**
This security function enforces the Firewall access control policy and the JCVM information flow control policy at runtime. It defines how accessing the following items: Static Class Fields, Array Objects, Class Instance Object Fields, Class Instance Object Methods, Standard Interface Methods, Shareable Interface Methods, Classes, Standard Interfaces, Shareable Interfaces, Array Object Methods.
Based on security attributes (Sharing, Context, Lifetime), it performs access control to object fields between objects and throws security exception when access is denied. Thus, it enforces applet isolation located in different packages and controls the access to global data containers shared by all applet instances.
The JCRE shall allocate and manage a context for each Java API package containing applets. The JCRE maintains for its own context a special system privilege so that it can perform operations that are denied to contexts of applets.

**JCS.Package**
This security function manages packages. A package is a structural item defined for naming, loading, storing, execution context definition. There are rules for package identification, for structure check and access rules definition. If inconsistent items are found during checks, an error message is sent.

**JCS.Crypto**
The security function offers the following services to applets thanks to the JavaCard API:

- Generation of random number as defined in [JCAPI301] and conformant to NIST SP800-90 to be used for key values or challenges during external exchanges,
- Computation of checksum CRC16 and CRC32 conformant with ISO3309, as defined in [JCAPI301],
- Ciphering and deciphering operation using TDES algorithm as defined in [JCAPI301],
- Generation of 4-byte or 8-byte MAC using TDES (112 or 168 bits key) algorithm according to [JCAPI301],
- Ciphering and deciphering operation using AES (128, 192 or 256 bits key) algorithm as defined in [JCAPI301],

- Generation of 16-byte, 24-byte or 32-byte MAC using AES algorithm in CBC mode with padding scheme (NOPAD), as defined in [JCAPI301],
- Data Hash operation for message digests using MD5 and SHA algorithms (SHA1, SHA224, SHA-256, SHA-384, SHA-512), as defined in [JCAPI301],
- HMAC generation based on MD5, SHA1, SHA256, SHA384 and SHA512 hash algorithms, as defined in [JCAPI301],
- Ciphering and deciphering operation using RSA with Standard and CRT algorithms, with padding scheme PKCS#1 or NOPAD, according to [JCAPI301],
- Generation and verification of RSA signatures in Standard or CRT modes, as defined in [JCAPI301]
- Generation and verification of ECDSA signatures, as defined in [JCAPI301]
- Generation of shared secrets according to the ECDH algorithm, as defined in [JCAPI301].

These operations are performed in a way to avoid revealing the key values. If the applet specifies an algorithm that the platform does not support, the JCRE refuses to perform the cryptographic operation and generates an exception. Even if [JCAPI301] specifies some other algorithms or parameters for cryptographic operations, the use of these other values are not advised; and clearly out of scope of the TOE. See [AGD] for details.

**JCS.RNG**
This security function provides random value using a given algorithm with or without a seed as defined in [JCAPI301].

**JCS.KeyManagement**
This security function enforces key management for the different associated operations: key building, key agreement, key generation, key importation, key exportation, key masking, key destruction using standard API defined in [JCAPI301].
Key generation supports the generation of RSA and ECDSA key pairs using a secure random number generator compliant with NIST SP800-90.
Key importation and exportation is done using method protecting confidentiality and integrity of key.
Key agreement enables an applet to agree on a shared secret with the external, with a method conformant to [JCAPI301]. It is built to avoid disclosure of this secret to third parties observing the exchange done for key agreement.
Key masking protects the confidentiality of cryptographic keys from being read out from the memory. It ensures the service of accessing and modifying them.
Key destruction disables the use of a key both logically and physically. Reuse is only possible after erase.

**JCS.OwnerPIN**
This security function supplies to applets a means to perform user identification and authentication with the OwnerPin class conformant to [JCAPI301].
It offers to create a PIN and store it securely in the persistent memory. It allows access to PIN value only to perform a secure comparison between a PIN stored in the persistent memory and a data received as parameter.
A method returns a positive result if a valid Pin has been presented during current session. If the PIN is not blocked and the comparison is successful, the validated flag is set to and the try counter is set to its maximum, otherwise the authentication fails and the associated try counter is decremented. When the validated flag is set, it is assumed that the user is authenticated.
When the try counter reaches zero, the PIN is blocked and the authentication is no more possible until the PIN is unblocked.

**JCS.EraseResidualData**
This security function ensures that sensitive data are locked upon the following operations as defined in [JCRE301]:

- Deletion of package and/or applications,
- Deletion of objects.

They are erased when space needs to be reused for allocation of new objects.
This security function also ensures that the sensitive temporary buffers (transient object, bArray object, APDU buffer, Cryptographic buffer) are securely cleared after their usage with respect to their life-cycle and interface as defined in [JCRE301], transient object at reset or allocation and persistent object are erased at allocation for new object.

**JCS.OutOfLifeDataUndisclosure**
This security function ensures that sensitive data are locked until postponed erasure on the following operations: Deletion of persistent and transient objects according to [JCRE301].

**JCS.RunTimeExecution**
This security function provides a secure run time environment and deals with:

- Instance registration or deletion,
- Application selection,
- Applet opcode execution,
- JCAPI methods execution,
- Logical channel management,
- APDU flow control, dispatch and buffer management,
- JCRE memory and context management,
- JCRE reference deletion,
- JCRE access rights,
- JCRE throw exception,
- JCRE security reaction.

**JCS.Exception**
This security function manages throwing of an instance of Exception class in the following cases:

- a SecurityException when an illegal access to an object is detected,
- a SystemException with an error code describing the error condition,
- a CryptoException in case of algorithm error or illegal use,
- any exception decided by the applet or the JCRE handled as temporary JCRE entry point object with associated JC API. It also offers a means to applet to handle exception and to JCRE to handle uncaught exception by applets.

**SA.FlowControl** *(Secure API)*
This security function provides means to applications to control execution flow, to detect any failure and to react if required.

**SA.SecureOperation** *(Secure API)*
This security function provides means to applications to securely perform data transfer and comparison, to detect any failure during operation and to react if required.

**SA.RandomDelay** *(Secure API)*
This security function provides means to introduce dummy operations leading to unobservability of sensitive operation.

**GA.OptionalServiceActivation** *(GemActivate)*
Activation is only possible for deactivated services defined in registry. Activation is done by changing internal state of optional platform service: cryptographic algorithm, package, applet instance, scalability (extension of available NVM) and NFC interface (SWP, HCI gate). It also allows activation of patch loaded on the platform. The command is available only for GemActivate under control of GemActivate Administrator. GemActivate is accessible only using a secure channel under MNO control.

**GA.PatchManagement** *(GemActivate)*
It manages patch identification when loaded. It allows activation of patch loaded on the platform and deactivation of activated patch by changing patch internal state. The command is available only for GemActivate under control of GemActivate Administrator. GemActivate is accessible only using a secure channel under MNO control.

**GA.ServiceAudit** (GemActivate)
It allows MNO or GemActivate administrator audit of actual state (deactivated, activated, inhibited) of each optional platform service described in platform registry.

**GA.GemActivateActivation** *(GemActivate)*
An application or a patch can be activated by GemActivate Administrator only if the following conditions are fulfilled:

- if patchability activation status is set,
- if activation command is consistent,
- if ratification counter limit is not reached,
- if anti-replay verification has not failed,
- if activation signature verification has not failed.

**GA.GemActivateAtomicActivation**
Patch activation and identification are atomic operations managed by GemActivate Administrator. Operations are completely fulfilled or cancelled in case of failure.

**OS.Atomicity** *(OS)*
This security function performs write operations atomically on complex type or object in order to avoid incomplete update. Prior to be written, data is stored in an atomic back-up area. In case on writing interrupt, the only two possible values are: initial value if writing is not started or final value if writing is started. At next start-up, the atomic back-up area is check to finalize interrupted writing.

**OS.MemoryManagement** *(OS)*
This security function allocates memory areas and performs access control on them to avoid unauthorized access. It manages circular writing to avoid instable memory state. It enforces memory recovery in case of error detection. It offers (when required) confidentiality services for data storage: Ciphering / Deciphering of Data in RAM or in FLASH, Scrambling / Unscrambling of Data in RAM or in FLASH.

**OS.PatchRegistry** *(OS)*
This security function manages modification of item registration in order to update the initial platform reference with patch reference to obtain the final platform reference.

**OS.PatchAtomicOperation** *(OS)*
This security function manages operations for patch loading, identification and activation in order to be in one of the following secure states: Patch is not loaded; Patch is loaded; Patch is loaded, identified and activated; Patch is unactivated and no more identified; Patch is deleted.

## 10.2. ST33G1M2 INTEGRATED CIRCUIT

**IC.LimitedFaultTolerance**
The TSF provides limited fault tolerance, by managing a certain number of faults or errors that may happen, related to memory contents, CPU, random number generation and cryptographic operations, thus preventing risk of malfunction. It is related to FRU_FLT.2 from [ST_ST33G1M2].

**IC.SecureState**
The TSF provides preservation of secure state by detecting and managing security violations, resulting in an immediate reset. It is related to FPT_FLS.1 from [ST_ST33G1M2].

**IC.LIM.Capability(TEST)**
The TSF ensures that only very limited test capabilities are available in USER configuration. It is related to FMT_LIM.1 [TEST] from [ST_ST33G1M2].

**IC.LIM.Capability(ADMIN)**
The TSF ensures that the Secure Flash Loader and the final test capabilities are unavailable in USER configuration. It is related to FMT_LIM.1 [ADMIN] from [ST_ST33G1M2].

**IC.ModeControl**
The TSF ensures that only defined modes are available: TEST, ADMIN, USER configuration. The TSF ensures the switching and the control of TOE configuration. The TSF reduces the available features depending on the TOE configuration. It is related to FMT_LIM.2 [TEST] & [ADMIN] from [ST_ST33G1M2].

**IC.AuditStorage**
The TSF provides command to store data for audit purpose using commands only available to authorized process. It is related to FAU_SAS.1 from [ST_ST33G1M2].

**IC.ResistanceToPhysicalAttack**
The TSF ensures resistance to physical tampering using features against probing and an active shield detecting integrity violation. It is related to FPT_PHP.3 from [ST_ST33G1M2].

**IC.InternalDataTransferProtection**
The TSF prevents disclosure of internal and user data thanks to memory scrambling and encryption, bus encryption… It is related to FDP_ITT.1, FPT_ITT.1, FDP_IFC.1 from [ST_ST33G1M2].

**IC.RandomNumberGeneration**
The TSF produces AIS31-qualified random numbers that can be directly used in embedded software. It is related to FCS_RNG.1 from [ST_ST33G1M2].

**IC.CryptoAccelerator**
The TSF provides EDES accelerator to perform DES and TDES encryption and decryption conformant to FIPS PUB 46-3. The TSF provides AES accelerator to perform AES encryption and decryption conformant to FIPS PUB 197. The TSF provides arithmetic primitives to be used in more complex computation in software cryptographic library. It is related to FCS_COP.1 [EDES] and FCS_COP.1 [AES] from [ST_ST33G1M2]. It also uses RNG, arithmetic primitives of Nescrypt. But there is no usage of NesLib.

**IC.MemoryProtection**
The TSF enforces a default memory protection policy when none other is programmed by the embedded software. It is related to FMT_MSA.3 [Memories] from [ST_ST33G1M2].

**IC.MPU**
The TSF provides a dynamic Memory protection unit (MPU) that can be configured by the ES. It is related to FMT_MSA.1 [Memories], FMT_SMF.1 [Memories] from [ST_ST33G1M2].

**IC.LoadingAccessControl**
The TSF provides an access control to loading. The Standard Loader instructions and/or Advanced Loader instructions can be executed only if valid passwords have been presented. It is related to FDP_ACC.1 [Loader], FDP_ACF.1 [Loader] from [ST_ST33G1M2].

## 10.3. TSS RATIONALE

| Security Functional Requirement | Coverage by TSS Security Function(s) |
|---|---|
| **FDP_UIT.1/CCM** | **GP.CardContentManagement**, **GP.SecurityDomain** and **GP.SecureChannel** manage CCM flow control |
| **FDP_ROL.1/CCM** | **GP.CardContentManagement**, **GP.ISD**, **GP.SSD, GP.VASD** and **GP.CASD** manage roles for respectively ISD, SSD, VASD and CASD, as well as associated delegated management |
| **FDP_ITC.2/CCM** | **GP.CardContentManagement** manages CCM Flow control |
| **FPT_FLS.1/CCM** | **GP.CardContentManagement** and **GP.SecurityDomain** manage failures when authentication fails and CCM fails. |
| **FCS_COP.1/DAP** | **GP.CardContentManagement** manages DAP verification |
| **FDP_ACC.1/SD** | **GP.SecurityDomain** manages access to Load File based on Delegation Token and DAP Block verification. **GP.GPRegistry** manages access and privileges to application code and instance using GP privileges. |
| **FDP_ACF.1/SD** | **GP.SecurityDomain** manages access to Load File based on Delegation Token and DAP Block verification. **GP.GPRegistry** manages access and privileges to application code and instance using GP privileges. |
| **FMT_MSA.1/SD** | **GP.SecurityDomain** allows modifying security attributes |
| **FMT_MSA.3/SD** | This SFR is fully covered by **GP.SecurityDomain** allowing definition of default values of security attributes |
| **FMT_SMF.1/SD** | **GP.SecurityDomain** and **GP.GPRegistry** allow modifying the behavior of security functions. |
| **FMT_SMR.1/SD** | **GP.SecurityDomain**, **GP.ISD**, **GP.SSD, GP.VASD** and **GP.CASD** manage the roles |
| **FTP_ITC.1/SC** | **GP.SecureChannel** manages secure channel and associated operations |
| **FCO_NRO.2/SC** | **GP.SecureChannel** manages secure channel |
| **FDP_IFC.2/SC** | **GP.SecureChannel** manages information flow control |
| **FDP_IFF.1/SC** | **GP.SecureChannel** manages information flow control |
| **FMT_MSA.1/SC** | **GP.SecureChannel** allows modifying security attributes |
| **FMT_MSA.3/SC** | **GP.SecureChannel** allows setting default values of security attributes |
| **FMT_SMF.1/SC** | **GP.SecurityDomain** and **GP.SecureChannel** allow role definition. |
| **FIA_UID.1/SC** | **GP.SecureChannel** manages mutual authentication and allowed operation prior identification |
| **FIA_UAU.1/SC** | **GP.SecureChannel** manages mutual authentication and allowed operation prior identification |
| **FIA_UAU.4/SC** | **GP.SecureChannel** manages mutual authentication with anti-replay mechanism. |
| **FDP_ACC.2/FIREWALL** | This SFR is fully covered by **JCS.Firewall** |
| **FDP_ACF.1/FIREWALL** | This SFR is fully covered by **JCS.Firewall** |
| **FDP_IFC.1/JCVM** | This SFR is fully covered by **JCS.Firewall** and **JCS.APDUBuffer** controlling unauthorized access or invalid storage of reference |
| **FDP_IFF.1/JCVM** | This SFR is fully covered by **JCS.Firewall** |
| **FDP_RIP.1/OBJECTS** | This SFR is fully covered by **JCS.OutOfLifeDataUndisclosure** (to avoid access to data prior erase) and **JCS.EraseResidualData** (to erase data) |
| **FMT_MSA.1/JCRE** | This SFR is fully covered by **JCS.RunTimeExecution** covering context switch and application selection |
| **FMT_MSA.1/JCVM** | This SFR is fully covered by **JCS.ByteCodeExecution** requiring context switch for specific code execution and **JCS.RunTimeExecution** covering context switch and modification of the Currently Active Context according to given rules |
| **FMT_MSA.2/FIREWALL_JCVM** | This SFR is fully covered by **JCS.RunTimeExecution** covering object sharing |

| Security Functional Requirement | Coverage by TSS Security Function(s) |
|---|---|
| **FMT_MSA.3/FIREWALL** | This SFR is fully covered by **JCS.RunTimeExecution** covering object sharing |
| **FMT_MSA.3/JCVM** | This SFR is fully covered by **JCS.RunTimeExecution** covering object sharing |
| **FMT_SMF.1** | This SFR is fully covered by **JCS.RunTimeExecution** covering context management and instance registration |
| **FMT_SMR.1** | This SFR is fully covered by **JCS.RunTimeExecution** covering JCVM and JCRE roles |
| **FCS_CKM.1/TDES** | This SFR is fully covered by **JCS.KeyManagement** covering key generation |
| **FCS_CKM.1/AES** | This SFR is fully covered by **JCS.KeyManagement** covering key generation |
| **FCS_CKM.1/RSA** | This SFR is fully covered by **JCS.KeyManagement** covering key generation |
| **FCS_CKM.1/ECDSA** | This SFR is fully covered by **JCS.KeyManagement** covering key generation |
| **FCS_CKM.1/HMAC** | This SFR is fully covered by **JCS.KeyManagement** covering key generation |
| **FCS_CKM.2/TDES** | This SFR is fully covered by **JCS.KeyManagement** covering key distribution |
| **FCS_CKM.2/AES** | This SFR is fully covered by **JCS.KeyManagement** covering key distribution |
| **FCS_CKM.2/RSA** | This SFR is fully covered by **JCS.KeyManagement** covering key distribution |
| **FCS_CKM.2/ECDSA** | This SFR is fully covered by **JCS.KeyManagement** covering key distribution |
| **FCS_CKM.2/HMAC** | This SFR is fully covered by **JCS.KeyManagement** covering key distribution |
| **FCS_CKM.3/TDES** | This SFR is fully covered by **JCS.KeyManagement** covering key access |
| **FCS_CKM.3/AES** | This SFR is fully covered by **JCS.KeyManagement** covering key access |
| **FCS_CKM.3/RSA** | This SFR is fully covered by **JCS.KeyManagement** covering key access |
| **FCS_CKM.3/ECDSA** | This SFR is fully covered by **JCS.KeyManagement** covering key access |
| **FCS_CKM.3/HMAC** | This SFR is fully covered by **JCS.KeyManagement** covering key access |
| **FCS_CKM.4** | This SFR is fully covered by **JCS.KeyManagement** covering key deletion |
| **FCS_COP.1/TDES_CIPHER** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/TDES_MAC** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/AES_CIPHER** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/AES_MAC** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/RSA_SIGN** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/RSA_CIPHER** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/ECDSA_SIGN** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/ECDH** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/HMAC** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FDP_RIP.1/ABORT** | This SFR is fully covered by **JCS.EraseResidualData** covering data erasure |
| **FDP_RIP.1/APDU** | This SFR is fully covered by **JCS.EraseResidualData** covering data erasure |
| **FDP_RIP.1/bArray** | This SFR is fully covered by **JCS.OutOfLifeDataUndisclosure** and **JCS.EraseResidualData** covering data erasure |
| **FDP_RIP.1/KEYS** | This SFR is fully covered by **JCS.EraseResidualData** covering data erasure |
| **FDP_RIP.1/TRANSIENT** | This SFR is covered by **JCS.OutOfLifeDataUndisclosure** managing the access control to transient object to be erased prior the erasure of the content in memory |
| **FDP_ROL.1/FIREWALL** | This SFR is fully covered by **JCS.RunTimeExecution** covering transaction rollback during specific operations |
| **FAU_ARP.1** | This SFR is fully covered by **JCS.RunTimeExecution**, **JCS.Exception**, **JCS.Firewall**, and **OS.MemoryManagement** covering exception handling with different specific operations |
| **FDP_SDI.2** | This SFR is fully covered by **JCS.OwnerPIN**, **JCS.KeyManagement**, **OS.Atomicity** and **OS.MemoryManagement** covering integrity handling with specific operations |
| **FPR_UNO.1** | This SFR is fully covered by **JCS.OwnerPIN, JCS.KeyManagement** and **OS.MemoryManagement** covering data handling with specific operations avoiding observation |
| **FPT_FLS.1/JCS** | This SFR is fully covered by **JCS.Exception**, **JCS.ByteCodeExecution**, **JCS.RunTimeExecution**, and **OS.Atomicity** preserving a secure state when unexpected events occur during specific operations |
| **FPT_TDC.1** | This SFR is fully covered by **JCS.Package** and **OS.MemoryManagement** assuming export check, CAP file translation and link specific operations |
| **FIA_ATD.1/AID** | This SFR is fully covered by **JCS.RunTimeExecution** and **GP.GPRegistry** controlling applet registration and uninstallation |
| **FIA_UID.2/AID** | This SFR is fully covered by **GP.GPRegistry** and **JCS.RunTimeExecution** managing user identity (package AID) during applet selection and identify associated context provided |
| **FIA_USB.1/AID** | This SFR is fully covered by **GP.GPRegistry** and **JCS.RunTimeExecution** managing registration of each applet and associated package during its installation with its AID |

| Security Functional Requirement | Coverage by TSS Security Function(s) |
|---|---|
| **FMT_MTD.1/JCRE** | This SFR is fully covered by **JCS.RunTimeExecution** offering services for applet registration and uninstallation managing associated access rights |
| **FMT_MTD.3/JCRE** | This SFR is fully covered by **JCS.RunTimeExecution** managing presence and legacy of AID with ISO rules |
| **FDP_ITC.2/Installer** | This SFR is fully covered by **JCS.Package** checking the binary compatibility of dependant packages using their version numbers and AIDs prior to installation operations |
| **FMT_SMR.1/Installer** | This SFR is fully covered by **JCS.RunTimeExecution**, **GP.SecurityDomain** covering the RTE, ISD and SSD roles |
| **FPT_FLS.1/Installer** | This SFR is fully covered by **JCS.Package**, **JCS.RunTimeExecution** and **GP.CardContentManagement** covering the applet instance registration operations and associated error handling |
| **FPT_RCV.3/Installer** | This SFR is fully covered by **JCS.RunTimeExecution**, **OS.MemoryManagement**, **GP.GPRegistry** and **GP.CardContentManagement** covering the applet instance erasure when applet instance registration operation fails |
| **FDP_ACC.2/ADEL** | This SFR is fully covered by **GP.CardContentManagement**, **GP.GPRegistry** and **JCS.RunTimeExecution** checking rules for applet instance uninstallation and deletion dependency rules |
| **FDP_ACF.1/ADEL** | This SFR is fully covered by **GP.CardContentManagement**, **GP.GPRegistry** and **JCS.RunTimeExecution** checking rules for applet instance uninstallation and deletion dependency rules |
| **FDP_RIP.1/ADEL** | This SFR is fully covered by **GP.CardContentManagement** and **JCS.OutOfLifeDataUndisclosure** by checking operations to avoid access to freed resources prior to its reuse |
| **FMT_MSA.1/ADEL** | This SFR is fully covered by **GP.GPRegistry**, **GP.CardContentManagement** and **JCS.RunTimeExecution** responsible of checking rules concerning applet attributes, implicit and explicit selection rules prior to authorize deletion operation |
| **FMT_MSA.3/ADEL** | This SFR is fully covered by **JCS.RunTimeExecution** and **GP.CardContentManagement** dealing with Security Attributes initialization, providing secure, restrictive default values for the security attributes of subject and objects involved in applet deletion |
| **FMT_SMF.1/ADEL** | This SFR is fully covered by **GP.CardContentManagement**, **GP.SecurityDomain** and **JCS.RunTimeExecution** supplying the following management functions: Modify the ActiveApplets security attribute |
| **FMT_SMR.1/ADEL** | This SFR is covered by **GP.SecurityDomain** maintaining the roles: (ISD & SDD) responsible of applet deletion. This SFR is also covered by **JCS.RunTimeExecution** maintaining the role (RTE) for applet uninstallation |
| **FPT_FLS.1/ADEL** | This SFR is covered by **GP.GPRegistry**, **JCS.RunTimeExecution** and **OS.Atomicity** preserving a secure state when unexpected events occur during package or instance deletion, managing the transaction part of the deletion operation by either rolling back, or completing it |
| **FDP_RIP.1/ODEL** | This SFR is covered by **JCS.EraseResidualData** and **OS.MemoryManagement** ensuring that the content of deleted objects is erased upon the deletion and by **JCS.OutOfLifeDataUndisclosure** making unavailable for disclosure upon further reallocation of the freed space |
| **FPT_FLS.1/ODEL** | This SFR is covered by **JCS.RunTimeExecution** and **OS.MemoryManagement** performing memory management to release no more used memory on unreferenced objects and preserves a secure state when unexpected events occur during object deletion |
| **FCO_NRO.2/CM** | This SFR is covered by **GP.SCP** managing the secure channel protocol where several checks are performed prior EF loading: * mutual authentication between the external entity (Issuer or Application provider) and the selected security Domain, including creation of a session key, * by the verification of a (chained) MAC that the Issuer or Application provider attaches to each file block sent, * by the erase of the session key at the end of the session. |
| **FDP_IFC.2/CM** | This SFR is covered by **GP.CardContentManagement** managing flow control between operations for loading, installing, selecting and executing application instances |
| **FDP_IFF.1/CM** | This SFR is covered by **GP.CardContentManagement** managing flow control between operations for loading, installing, selecting and executing application instances |
| **FDP_UIT.1/CM** | This SFR is covered by **GP.SCP** providing a session key generation. It ensures |

| Security Functional Requirement | Coverage by TSS Security Function(s) |
|---|---|
| | that the whole package has been correctly received |
| **FIA_UID.1/CM** | This SFR is covered by **JCS.RunTimeExecution** and **GP.SecurityDomain** controlling accessible action prior identification and action when SD or application associated to SD are selected |
| **FMT_MSA.1/CM** | This SFR is covered by **GP.SCP** setting of command security level at initialization and checking command security level during execution |
| **FMT_MSA.3/CM** | This SFR is covered by **GP.SCP** providing setting of the default value |
| **FMT_SMF.1/CM** | This SFR is covered by **GP.SCP** by setting the security level of the Secure Channel as requested by the authenticated external entity (Issuer or application provider) and updating the current value of the ICV upon reception of a new message through the Secure Channel |
| **FMT_SMR.1/CM** | This SFR is covered by **GP.SecurityDomain**, **GP.ISD**, **GP.SSD, GP.VASD** and **GP.CASD** managing the roles: issuer, application provider, verification authority and controlling authority |
| **FTP_ITC.1/CM** | This SFR is covered by **GP.SCP** for applet loading |
| **FPT_RCV.3/OS** | This SFR is covered by **OS.Atomicity** |
| **FPT_RCV.4/OS** | This SFR is covered by **OS.MemoryManagement** |
| **FCS_COP.1/Hash** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_COP.1/CRC** | This SFR is fully covered by **JCS.Crypto** covering cryptographic operation |
| **FCS_RND.1** | This SFR is covered by **JCS.RNG** providing a dedicated API to applet. **JCS.RNG** uses **IC.RandomNumberGeneration** to supply the service |
| **FPT_FLS.1/SecureAPI** | This SFR is fully covered by **SA.SecureOperation**, **SA.FlowControl** and **IC.LimitedFaultTolerance** |
| **FPT_ITT.1/SecureAPI** | This SFR is fully covered by **SA.SecureOperation** and **IC.InternalDataTransferProtection** |
| **FPR_UNO.1/SecureAPI** | This SFR is fully covered by **SA.RandomDelay**, **SA.SecureOperation** and **IC.RandomNumberGeneration** |
| **FMT_SMR.1/GemActivate** | This SFR is fully covered by **GA.GemActivateActivation** maintaining GemActivate Administrator role |
| **FMT_SMF.1/GemActivate** | This SFR is fully covered by **GA.GemActivateActivation** |
| **FMT_MOF.1/GemActivate** | This SFR is fully covered by **GA.OptionalServiceActivation** |
| **FMT_MSA.1/GemActivate** | This SFR is fully covered by **GA.OptionalServiceActivation** |
| **FMT_MTD.1/GemActivate** | This SFR is fully covered by **GA.ServiceAudit** |
| **FDP_ACC.1/GemActivate_DAP** | This SFR is fully covered by **GA.GemActivateActivation** |
| **FDP_ACF.1/ GemActivate_DAP** | This SFR is fully covered by **GA.GemActivateActivation** |
| **FMT_MSA.3/GemActivate_DAP** | This SFR is fully covered by **GA.GemActivateActivation** |
| **FIA_ATD.1/AID_Patch** | This SFR is fully covered by **JCS.RunTimeExecution**, **GP.PATCH_IDENTIFICATION**, **GP.GPRegistry**, **GA.PatchManagement, GA.GemActivateAtomicActivation, OS.PatchRegistry** and **OS.PatchAtomicOperation** controlling patch registration and uninstallation. |

**END OF DOCUMENT**