	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Security Target

**Formal assurances
on the
Java Card Virtual Machine
of
LinqUs USIM 128k PK certified
using SC33F640**



	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

Table of Contents

1	ST INTRODUCTION	5
1.1	ST REFERENCE	5
1.2	TOE REFERENCE	6
1.3	TOE OVERVIEW	6
1.3.1	<i>TOE Type</i>	6
1.3.2	<i>TOE usage</i>	7
1.3.3	<i>TOE Boundaries</i>	8
1.3.4	<i>TOE Description</i>	9
1.3.5	<i>TOE Life Cycle</i>	10
1.3.6	<i>TOE Environment</i>	12
1.3.7	<i>Actors of the TOE</i>	14
1.3.8	<i>TOE Security Features</i>	14
1.3.9	<i>Non-TOE HW/SW/FW Available to the TOE</i>	16
2	CONFORMANCE CLAIMS	17
2.1	CC CONFORMANCE CLAIMS	17
2.2	PP CONFORMANCE CLAIMS	17
2.3	CONFORMANCE RATIONALE	17
2.3.1	<i>PP USIM</i>	17
2.3.2	<i>PP JCS</i>	17
3	SECURITY PROBLEM DEFINITION	22
3.1	ASSETS	22
3.1.1	<i>Java Card System Protection Profile - Open Configuration</i>	22
3.2	THREATS	23
3.2.1	<i>Java Card System Protection Profile - Open Configuration</i>	23
3.3	ORGANIZATIONAL SECURITY POLICIES	27
3.3.1		27
3.4	ASSUMPTIONS	27
3.4.1	<i>Java Card System Protection Profile - Open Configuration</i>	27
3.4.2	<i>Additional assumptions</i>	27
4	SECURITY OBJECTIVES	30
4.1	SECURITY OBJECTIVES FOR THE TOE	30
4.1.1	<i>Java Card System Protection Profile - Open Configuration</i>	30
4.2	SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT	32
4.2.1	<i>Java Card System Protection Profile - Open Configuration</i>	32
4.2.2	<i>Additional security objectives</i>	33
4.3	SECURITY OBJECTIVES RATIONALE	35
4.3.1	<i>Threats</i>	35
4.3.2	<i>Organizational Security Policies</i>	41
4.3.3	<i>Assumptions</i>	41
4.3.4	<i>SPD and Security Objectives</i>	42

	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

5	SECURITY REQUIREMENTS	46
5.1	SECURITY FUNCTIONAL REQUIREMENTS	46
5.1.1	<i>Java Card System Protection Profile - Open Configuration.....</i>	<i>46</i>
5.2	SECURITY ASSURANCE REQUIREMENTS.....	61
5.3	SECURITY REQUIREMENTS RATIONALE	61
5.3.1	<i>Objectives</i>	<i>61</i>
5.3.2	<i>Rationale tables of Security Objectives and SFRs.....</i>	<i>63</i>
5.3.3	<i>Dependencies</i>	<i>64</i>
5.3.4	<i>Rationale for the Security Assurance Requirements</i>	<i>66</i>
6	TOE SUMMARY SPECIFICATION	71
6.1	TOE SUMMARY SPECIFICATION.....	71
6.2	SFRs AND TSS.....	72
6.2.1	<i>SFRs and TSS - Rationale</i>	<i>72</i>
6.2.2	<i>Association tables of SFRs and TSS</i>	<i>72</i>
7	NOTICE	73
8	REFERENCES, GLOSSARY AND ABBREVIATIONS	74
8.1	EXTERNAL REFERENCES.....	74
8.2	INTERNAL REFERENCES	75
8.3	ABBREVIATIONS.....	76
8.4	GLOSSARY	77



	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

Table of Figures

Figure 1: LinqUs USIM 128k card to be inserted in a mobile.....	7
Figure 2: TOE Physical Boundaries	8
Figure 3: TOE Logical Boundaries	8
Figure 4: TOE Life Cycle Refined	11

Table of Tables

Table 1 ST References	5
Table 2 TOE References.....	6
Table 3 Refinement of SFR of PP JCS.....	21
Table 4 Compatibility study	21
Table 5 Threats and Security Objectives - Coverage	43
Table 6 Security Objectives and Threats - Coverage	44
Table 7 OSPs and Security Objectives - Coverage.....	45
Table 8 Security Objectives and SFRs - Coverage	63
Table 9 SFRs and Security Objectives	64
Table 10 SFRs dependencies.....	65
Table 11 SARs dependencies.....	66

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

1 ST Introduction

1.1 ST Reference


Security Target and associated evaluation are completely defined by information located in the following table.

Title:	Security Target : Formal assurance on the Java card Virtual Machine of LinqUs USIM 128k using SC33F640
Reference:	D1185035
Version	1.5p
Origin:	GEMALTO
ITSEF:	THALES CEACI
Certification Body:	ANSSI
Evaluation scheme:	French

Table 1 ST References

This Security Target describes:

- The Target of Evaluation, the TOE components, the components in the TOE environment, the product type, the TOE environment and life cycle, the limits of the TOE.
- The assets to be protected, the threats to be countered by the TOE itself during the usage of the TOE
- The organizational security policies, and the assumptions,
- The security objectives for the TOE and its environment,
- The security functional requirements for the TOE and its IT environment,
- The TOE security assurance requirements
- The security functions and associated rationales.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

1.2 TOE Reference

Product and TOE are completely defined by information located in Table 3.

Product Name	LinqUs USIM 128k certified
Product Reference	T1017287
Product Version	Release A
TOE name	Java Card Virtual Machine of LinqUs USIM 128k platform using SC33F640
TOE Reference	S1092122
TOE Version	Release A
Commercial Name	LinqUs USIM 128k PK certified

Table 2 TOE References

1.3 TOE overview

1.3.1 TOE Type

The product **LinqUs USIM 128k certified** (also named **LinqUs USIM 128k**) is (U)SIM smart card defined to be used mainly in a mobile or a Smartphone, but can be used in any device with an interface conformant to [ISO 7816] specification. It is delivered using an ISO form factor including a plug-in form factor as defined in [TS 102 221] or 3FF form factor.


The product **LinqUs USIM 128k** implements the standard communication protocol (ISO 7816 T=0) and ETSI standard allowing communication between smartcard, mobile and server using OTA.

The Target of Evaluation (TOE) is the **Java Card Virtual Machine** embedded in the LinqUs USIM 128K that is an USIM card is intended to be plugged in a mobile phone or other mobile devices to provide services to an end user.

This security target defines the requirements of the Java Card Virtual Machine as a subset of the Java Card System, and corresponds to an extension of the evaluation of the *full* TOE of the product, described in the security target **LinqUs USIM 128k PK certified using SC33F640** [ST_linqUs128]. This security target restricts the security target [ST_LinqUs128] to the virtual machine, in charge of the secure execution of the applications after their loading on the USIM card.

More precisely, the TOE in this security target is made of:

- The linker
- The interpreter

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

- A (native) subset of the JC API

The TOE is a subset of the full TOE whose the configuration is defined using [PP-JCS] Java Card System protection profile Open Configuration.

1.3.2 TOE usage



Figure 1: LinqUs USIM 128k card to be inserted in a mobile

The present TOE, the Java Card Virtual Machine, is a subset of the product **LinqUs USIM 128K** that we describe the usage in this section.


The USIM defined in the [3GPP] standards as the Universal Subscriber Identity Module is an evolution of the SIM developed to ensure compliance within UMTS networks. A Subscriber Identity Module (SIM) is a removable module to plug within GSM mobile equipment that contains the International Mobile Subscriber Identity (IMSI) which unambiguously identifies a subscriber. It also stores other subscriber-related information or applications such as SIM Toolkit, and other application (as an E-sign application). In the rest of the document, the term of (U)SIM is used to refer to SIM or USIM as there are considered in the same way regarding security.

The primary services of the (U)SIM (when it is plugged in handset) are the user authentication by PIN capture and the SIM authentication on the MNO network, giving access to MNO services through the mobile. It also stores other subscriber-related information or applications such as SIM Toolkit applications as specified in [TS102.223] and [TS131.111].

The LinqUs USIM 128k Platform implements major industry standards:

- Java Card 2.2.2,
- Global Platform 2.2,
- Full ETSI release 6,
- 3GPP Release 6.

It supports **multiple networks (2G, 3G ...)** and it implies that several Network Access Applications (NAA) working together, requiring for dynamic switching from networks (3G to 2G, 2G to 3G). Each application is designed like a plug-in.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

1.3.3 TOE Boundaries

The following figures illustrate the TOE physical and logical boundaries.

The product is a smartcard including a plastic card and a module performing the interface between reader and the mobile and the embedded chip. The Target of Evaluation (TOE) is the Java Card Virtual Machine that is embedded in Smart Card Integrated Circuit in operation and in accordance to its functional specifications. Other smart card product items (such as plastic, module, bounding, printing...) and other embedded software (OS, Secure API, etc) are outside the scope of this evaluation

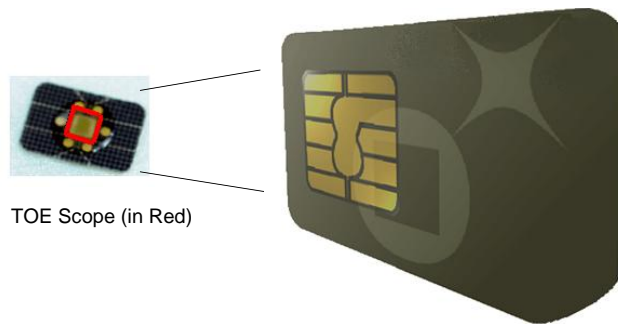


Figure 2: TOE Physical Boundaries

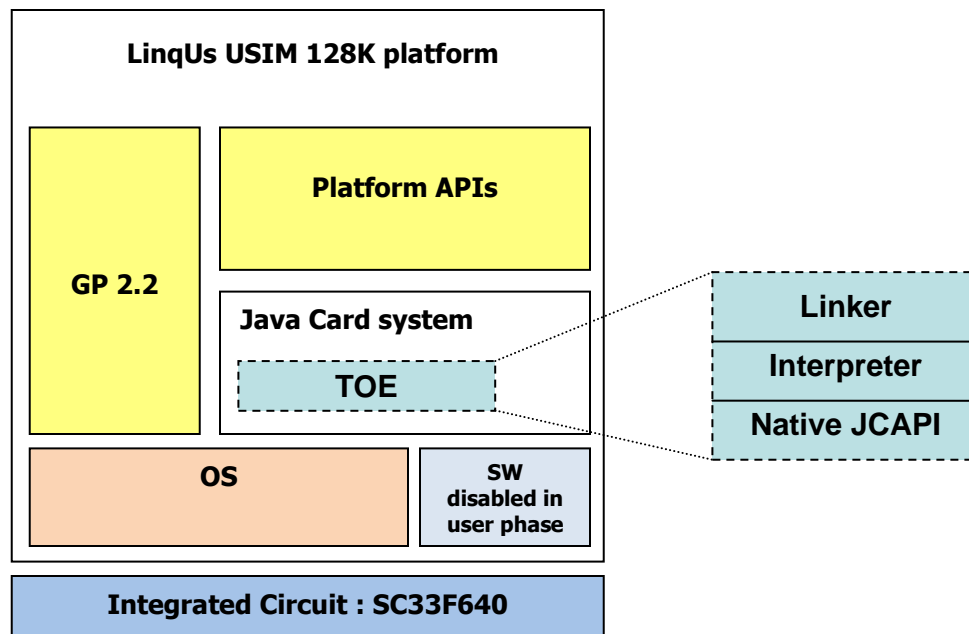



Figure 3: TOE Logical Boundaries

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

1.3.4 TOE Description

The present TOE is a subset (identified by the dotted lines in Figure 3: TOE Logical Boundaries) of the full TOE evaluated in LinqUS USIM 128k [ST_LinqUs128]. This subset ensures the **secure execution of an applet that has been byte code verified and loaded on the product**. This execution is processed in two phases:

Phase 1: the (static) link of the loaded CAP file (done once)

Phase 2: the (dynamic) interpretation of the linked byte-codes (done as many as necessary)

The TOE is composed of the following components (part of Java Card System):

- The **linker** (used for CAP file preparing for interpretation),
- The **interpreter** (used for bytecode interpreting), and
- The **firewall-related native Java Card API** (to ensure that it provides no means to bypass the firewall access control)

All these components are supplied by Gemalto and have the same version as the embedded software evaluated in LinqUS USIM 128k [ST_LinqUs128].

1.3.4.1 The linker

The linker is in charge of the rearrangement of the data structures contained in the Converted APplet (CAP) file in order to speed up the execution of the applet. The linker first performs a resolution step that is, resolves the external and internal references of a CAP file and replaces them by direct ones. Then it performs the preparation step, allocating the static field image and the static arrays. The later ones are also initialized, thus giving rise to the configuration that will constitute the corresponding initial state of the (JC) interpreter.

The linker contributes to the installation of post-issuance applets. The linker is invoked after the loading process by the JCRE to link the CAP file using the existing packages. Then, the API method `Applet.install` is invoked to instantiate the new application using a fresh AID. The other application management functionalities, such as the load process, the CAD communication, are out of the scope of the TOE.


During its lifetime, an applet can be updated by new packages to be loaded on the TOE. These (Java) updates may replace part of the original applet or extend it. The Java updates are also ensured by the linker.

1.3.4.2 The interpreter

Once an application is installed, registered and selected, its execution is carried out by the embedded interpreter. The interpreter mainly consists of a loop that computes the next bytecode to be executed and dispatch the appropriate interpretation functions. Such function modifies the runtimes data areas of the JVM (the heap, the static field images, the frame stack, etc) according to the semantics of the byte code interpreted.

1.3.4.3 The native Java Card API

The byte code interpretation done by the interpreter depends in turn on the behavior of methods of the API. The Java Card APIs consists of a set of customized classes for programming smart card applications according to the ISO 7816 model.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

Native API methods are usually written in C and are considered as parts of the Java Card platform. The native methods participate in enforcing several essential TOE security features such as firewall. Consequently, the native methods of the package `javacard.framework` are then included in the TOE. The package `javacard.framework.service` that is mainly used for the JCRMI functionality is not included.

The packages `javacardx.crypto` and `javacard.security` are not included as they are beyond the current state of the art of the formal model.

1.3.5 TOE Life Cycle


As the life cycle of the JC system, part of the USIM platform software, is the lifetime of the card, the life cycle is the product life cycle described in the [ST_LinqUs128].

Product life cycle is described in the following picture using [PP-USIM] description refined with Gemalto specific environment due to embedded software loading in flash in phase 6.

The (U)SIM platform life cycle is composed of four stages (as defined in PP (U)SIM figure 3):

- Development (embedded software and IC separately),
- Storage, pre-personalization and test,
- Loading, Personalization and test,
- Final usage.

Refined life cycle based on [PP-BSI-0035] with Gemalto product constraints is described in the Figure 4: TOE Life Cycle Refined.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

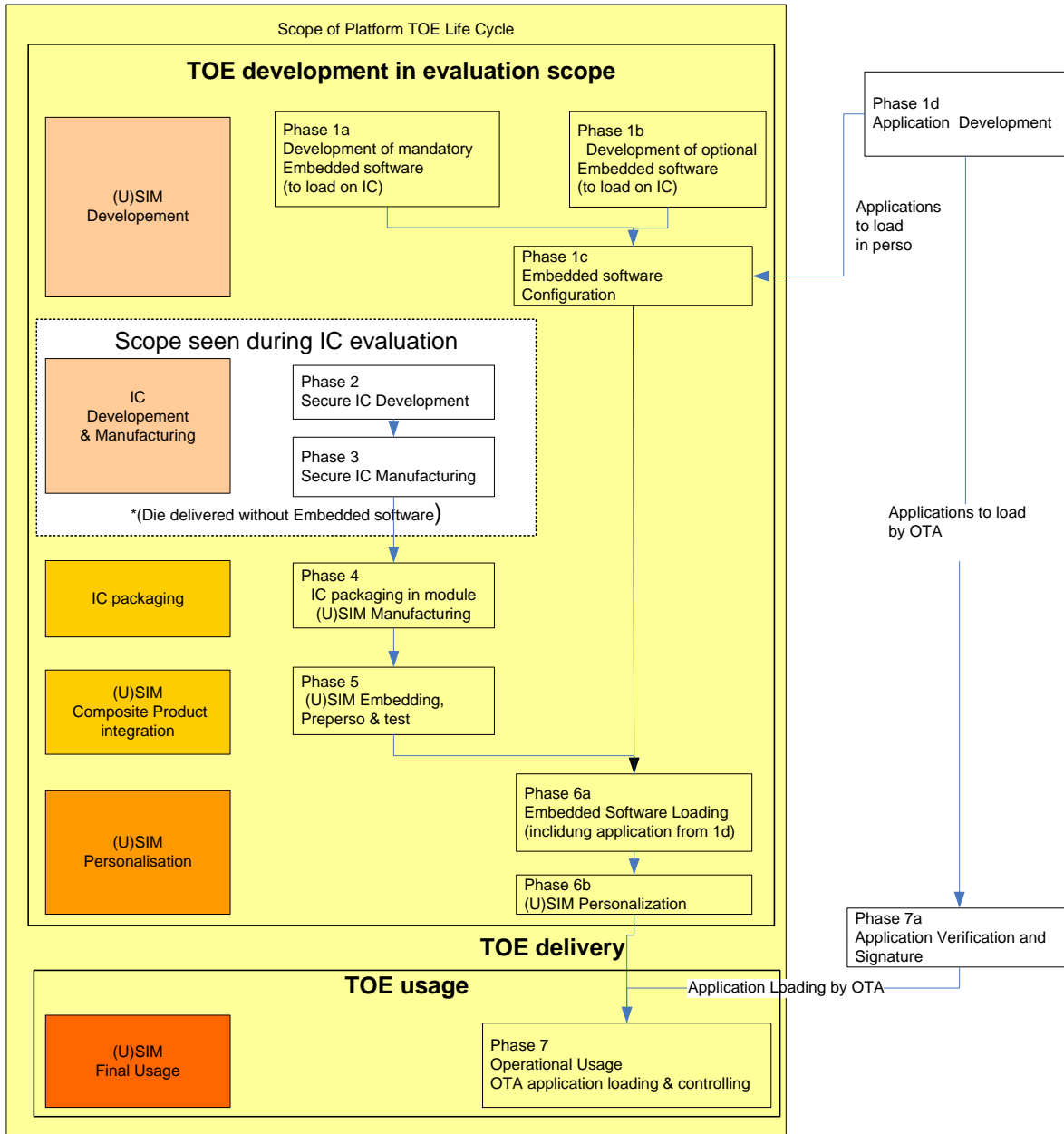



Figure 4: TOE Life Cycle Refined

The following phases corresponding to the one previously described are:

- Phases 1(a, b) correspond to the development of the TOE embedded software and its configuration (1c) with applications to be loaded in phase 6.
- Phase 2, 3 and 4 correspond to IC development, manufacturing and packaging in module, respectively.
- Phase 5 concerns the composite product integration with the module and other smart card items,

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

- Phase 6 (a, b) is dedicated to the TOE embedded software loading and product personalization prior to TOE delivery.
- Phase 7 is the product operational phase including application loading and controlling by VA authority.

Note: The IC used in the current life cycle does not contain any embedded software prior to phase 6. It is under protection of software security function of IC dedicated software. As generic product, the ICs are stored in personalization environment but there are not dedicated to the TOE. After loading in phase 6, IC loading service is locked and no more available after this phase (rf. FMT_LIM from [ST_IC]).

The TOE of this security target is a subset of the embedded software evaluated in [ST_LinqUs128] and is the product at the end of phase 6 as shown in previous figure. It is the operational LinqUs USIM 128k product, as a personalized smart card.

As far as the EAL4+ evaluation scope is concerned, phases 1 to 6 are considered as development and manufacturing phases of the product but the TOE is the result of these phases that can consequently be seen as phases of the TOE generation.

The TOE, subset of the USIM platform, is not a product but a subset of the software embedded into the LinqUs USIM 128K product. Then the TOE delivery is the product delivery performed at end of phase 6 and phase 7 is the operational phase of the TOE.

Out of the TOE evaluation scope, there are also the following operations linked to the TOE:

- in phase 1(d), the application development,
- in phase 7(a), the application verification and signature prior to application loading.


1.3.6 TOE Environment

Considering the TOE, the environment is defined as follows:

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

1.3.6.1 TOE Development Environment & Roles

The TOE described in this ST is developed in different places under the control of a defined administrator as indicated below:

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Phase	Administrator and Location
IC design and Dedicated Software development	STMicroelectronics Sites are defined in [ST/IC]
Embedded software Development	Gemalto (Meudon, La Ciotat, Singapore) Trusted Labs (Versailles)
Embedded software Configuration	Gemalto (Tczew, Gemenos)

1.3.6.2 TOE Manufacturing Environment

The TOE described in this ST is produced in different places under the control of a defined administrator as indicated below:

Phase	Administrator and Location
IC manufacturing and Testing	STMicroelectronics Sites are defined in [ST/IC]
IC packaging	Gemalto (Pont Audemer, Tczew)
Composite Product integration	Gemalto (Pont Audemer, Tczew)


1.3.6.3 TOE Personalization Environment

The TOE described in this ST is personalized in different places under the control of a defined administrator as indicated below:

Phase	Administrator and Location
Personalization	Gemalto (Pont Audemer, Tczew)
Delivery to Final user (MNO)	From Personalization site to MNO site

1.3.6.4 TOE User Environment

Smart Cards are used in a wide range of applications to assure authorized conditional access. This specific product is to be used on terminals such as GSM and UMTS handsets or smart card readers. The end-user environment therefore covers an unprotected environment, thus making it difficult to avoid any abuse of the TOE. The product is prepared accordingly to mitigate such attacks in this environment.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

The product is nevertheless under the control of the MNO administration using the OTA channel. The TOE can be blocked by GP administrative commands under administrator control.

Considering the TOE as the Java Card System, it is fully operational in the user environment. The only abuse that we concentrate on in this Security Target is the **applet isolation property**: the main requirement that has to be full filled is that each applet is protected from illegal modification from the neighborhood applet and that the underlying system is protected from fraudulent and corrupted applet.

1.3.7 Actors of the TOE

One of the characteristics of the (U)SIM Java Card platforms is that several entities are represented inside these platforms:

- The Mobile Network Operator (MNO or mobile operator), issuer of the (U)SIM Java Card platform and proprietary of the TOE. The TOE guarantees that the issuer, once authenticated, could manage the loading, instantiation or deletion of applications.
- The Application Provider (AP), entity or institution responsible for the applications and their associated services. It is a financial institution (a bank), a transport operator or a third party operator.
- The Controlling Authority (CA), entity independent from the MNO represented on the (U)SIM card and responsible for securing the keys creation and personalization of the Application Provider Security Domain (APSD) (Push and Pull personalization model of [GP-UICC]).
- The Verification Authority (VA), trusted third party represented on the (U)SIM card, acting on behalf of the MNO and responsible for the verification of applications signatures (mandated DAP) during the loading process. These applications shall be validated for the basic applications or certified for the secure ones.


1.3.8 TOE Security Features

The TOE can manage secure or basic applets. These applets can be loaded and instantiated onto the TOE either before card issuance or over-the-air (OTA) in post-issuance through the mobile network, without physical manipulation of the TOE and in a connected environment. Other administrative operations can also be done using OTA.

The main security feature of the TOE is the correct and secure execution of applications, in a connected environment and with the presence on the product of other basic applications.

While the Java Card virtual machine (JCVM) is responsible for ensuring language-level security, the JCRE provides additional security features for the product. The basic runtime security feature imposed by the JCRE enforces isolation of applets using the Java Card **firewall**. It prevents objects created by one applet from being used by another applet without explicit sharing. This prevents unauthorized access to the fields and methods of class instances, as well as the length and contents of arrays.

The firewall is the most important security feature of the Java Card System. It enables complete isolation between applets or controlled communication through additional mechanisms that allow them to share objects when needed. The JCRE allows such sharing using the concept of "shareable interface objects"

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

(SIO) and static public variables. The JCVM should ensure that the only way for applets to access any resources are either through the JCRE or through the native API.

1.3.8.1 Security services to applications

Among the security services provided by the USIM platform to the applications to protect their data and assets, the TOE of this security target is in charge of:

- Confidentiality and integrity of application data among applications. Applications belonging to different contexts are isolated from each other. Application data are not accessible by a normal or abnormal execution of another basic or secure application.
- Application code execution integrity. The Java Card VM and the "applications isolation" property guarantee that the application code is operating as specified in absence of perturbations.


The other security services ensured by the USIM product and evaluated during the LinqUs USIM 128k evaluation [ST_LinqUs128] are:

- Confidentiality and integrity of cryptographic keys and associated operations. Cryptographic operations are protected, including protection against observation or perturbation attacks. Confidentiality and integrity of cryptographic keys, application data are guaranteed at all time during execution of cryptographic operations.
- Confidentiality and integrity of authentication data. Authentication data are protected, including protection against observation or perturbation attacks. Confidentiality and integrity of authentication data, application data are guaranteed at all time during execution of authentication operations.
- Application code execution integrity in case of perturbation.

1.3.8.2 Application Management

The application management uses security services that are not in the scope of this TOE but are provided by the GlobalPlatform framework:

- The MNO as Card issuer is initially the only entity authorized to manage applications (loading, instantiation, deletion) through a secure communication channel with the card, based on SMS or BIP technology. However, the MNO can grant these privileges to the AP through the delegated management functionality of GP.
- Before loading, all applications are verified by a validation laboratory for the basic applications, or by an ITSEF for the secure applications. All loaded applications are associated at load time to a Verification Authority (VA) signature (Mandated DAP) that is verified on card by the on-card representative of the VA prior to the completion of the application loading operation and prior to the instantiation of any applet defined in the loaded application.
- Application Providers personalize their applications and Security Domains (APSD) in a confidential manner. Application Providers have Security Domain keysets enabling them to be authenticated to the corresponding Security Domain and to establish a trusted channel between the TOE and an external trusted device. These Security Domains keysets are not known by the Card issuer.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

Basic and Secure applets (as defined below) are loaded in different Java Card packages.


1.3.9 *Non-TOE HW/SW/FW Available to the TOE*

The TOE does not include the following components (that are part of Java Card System):

- Applet selection/deletion/loading
- Object deletion
- Java Card RMI
- Cryptographic API

The other non TOE HW/SW/FW are those defined in [PP JCS] and [PP-USIM] as:

- Byte Code verifier, Verification tool and Application DAP creation tool available to Verification authority, mobile handset, Terminal in point of sale, Remote server for administration and Trusted network and IT system for communication.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

2 Conformance claims

2.1 CC Conformance Claims

This Security Target has been written using CC version V3.1 release 3.

All the security requirements have been drawn from the catalogue of requirements in Part 2 [CC-2].

This Security Target is conformant with CC part 3 [CC-3].

The evaluation is performed according [CEM] and supporting documents [JIL] [Note/12.1][AIS34].

The assurance requirement of this security target is **EAL4 augmented by:**

- **ADV_SPM.1** Formal TOE security Policy Model
- **ADV_FSP.6** Complete semi-formal functional specification with additional formal specification.
- **ADV_TDS.6** Complete semi-formal modular design with high-level design presentation.
- **ADV_INT.3** Minimally complex internals
- **ADV_IMP.2** Complete mapping of the implementation representation of the TSF

Other augmentations are covered by the EAL4+ evaluation of the product and result from compliance to [PP-JCS]:

- **ALC_DVS.2** Sufficiency of security measures,
- **AVA_VAN.5** Advanced methodical vulnerability analysis.

2.2 PP Conformance claims

This Security Target does not claim any protection profile but it is based on the [PP-JCS] "open configuration". Also the TOE is part of the embedded software of the product LinqUs USIM 128K evaluated in the ST [ST_LinqUs128] that has a "demonstrable" conformance to [PP-JCS] "open configuration".

The TOE uses an Integrated Circuit certified with CC EAL5 augmented with ALC_DVS.2 and AVA_VAN.5.

The SC33F640 Security Target [ST/IC] claims strict conformance to the Security IC Platform Protection Profile [BSI-PP-2007-0035], as required by this Protection Profile.


Refinements of [BSI-PP-2007-0035] are described in [ST/IC] and are not repeated here.

2.3 Conformance rationale

2.3.1 PP USIM

The elements from [PP-USIM] are not described in the Security Problem Definition because none of them is in the scope of this TOE. These elements are dedicated in the ST [ST_LinqUs128] of the product LinqUs USIM 128K.


2.3.2 PP JCS

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77


The differences between this Security Target and the [PP-JCS] are described here.

Because the TOE is a subset of the reference TOE defined in [PP-JCS], only a subset of PP JCS SFRs are enforced in this evaluation. Table 3 Refinement of SFR of PP JCS explains how the SFRs of PP JCS are refined and used in this ST.


Consequently, only a subset of the security objectives defined in PP JCS are satisfied in this ST (because of the limited TOE security functions). The other objectives are put in the environment. Table 4 resumes the modifications done by this ST with respect to the PP JCS.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Functional requirements	Refined in [PP-JCS]	Refined in this ST
FDP_ACC.2/FIREWALL	Yes	No
FDP_ACF.1/FIREWALL	Yes	Yes
FDP_IFC.1/JCVM	Yes	No
FDP_IFF.1/JCVM	Yes	Yes
FDP_RIP.1/OBJECTS	Yes	Not used
FMT_MSA.1/JCRE	Yes	Not used
FMT_MSA.1/JCVM	Yes	No
FMT_MSA.2/FIREWALL_JCVM	Yes	No
FMT_MSA.3/FIREWALL	Yes	No
FMT_MSA.3/JCVM	Yes	No
FMT_SMF.1	Yes	No
FMT_SMR.1	Yes	No
FCS_CKM.1	Yes	Not used
FCS_CKM.2	Yes	Not used
FCS_CKM.3	Yes	Not used
FCS_CKM.4	Yes	Not used
FCS_COP.1	Yes	Not used
FDP_RIP.1/ABORT	Yes	Not used
FDP_RIP.1/APDU	Yes	Not used
FDP_RIP.1/bArray	Yes	Not used
FDP_RIP.1/KEYS	Yes	Not used
FDP_RIP.1/TRANSIENT	Yes	Not used
FDP_ROL.1/FIREWALL	Yes	No
FAU_ARP.1	Yes	Yes
FDP_SDI.2	Yes	Not used
FPR_UNO.1	Yes	Not used
FPT_FLS.1	Yes	No
FPT_TDC.1	Yes	No
FIA_ATD.1/AID	Yes	No
FIA_UID.2/AID	Yes	Not used

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Functional requirements	Refined in [PP-JCS]	Refined in this ST
FIA_USB.1/AID	Yes	Not used
FMT_MTD.1/JCRE	Yes	No
FMT_MTD.3/JCRE	Yes	No
FDP_ITC.2/Installer	Yes	Not used
FMT_SMR.1/Installer	Yes	Not used
FPT_FLS.1/Installer	Yes	Yes
FPT_RCV.3/Installer	Yes	Yes
FDP_ACC.2/ADEL	Yes	Not used
FDP_ACF.1/ADEL	Yes	Not used
FDP_RIP.1/ADEL	Yes	Not used
FMT_MSA.1/ADEL	Yes	Not used
FMT_MSA.3/ADEL	Yes	Not used
FMT_SMF.1/ADEL	Yes	Not used
FMT_SMR.1/ADEL	Yes	Not used
FPT_FLS.1/ADEL	Yes	Not used
FDP_ACC.2/JCRMI	Yes	Not used
FDP_ACF.1/JCRMI	Yes	Not used
FDP_IFC.1/JCRMI	Yes	Not used
FDP_IFF.1/JCRMI	Yes	Not used
FMT_MSA.1/EXPORT	Yes	Not used
FMT_MSA.1/REM_REFS	Yes	Not used
FMT_MSA.3/JCRMI	Yes	Not used
FMT_REV.1/JCRMI	Yes	Not used
FMT_SMF.1/JCRMI	Yes	Not used
FMT_SMR.1/JCRMI	Yes	Not used
FDP_RIP.1/ODEL	Yes	Not used
FPT_FLS.1/ODEL	Yes	Not used
FCO_NRO.2/CM	Yes	Not used
FDP_IFC.2/CM	Yes	Not used
FDP_IFF.1/CM	Yes	Not used


	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Functional requirements	Refined in [PP-JCS]	Refined in this ST
FDP_UIT.1/CM	Yes	Not used
FIA_UID.1/CM	Yes	Not used
FMT_MSA.1/CM	Yes	Not used
FMT_MSA.3/CM	Yes	Not used
FMT_SMF.1/CM	Yes	Not used
FMT_SMR.1/CM	Yes	Not used
FTP_ITC.1/CM	Yes	Not used

Table 3 Refinement of SFR of PP JCS

PP JCS elements	Modification in ST
Assets	Not changed
Threats	Not changed
Assumptions	Augmented
OSP	Not changed
Security objectives	Reduced
Security objective for the operational environment	Augmented
Security functional requirements	Reduced
Security assurance requirements	Augmented

Table 4 Compatibility study

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

3 Security problem definition

3.1 Assets

3.1.1 Java Card System Protection Profile - Open Configuration

Assets are security-relevant elements to be directly protected by the TOE. Confidentiality of assets is always intended with respect to un-trusted people or software, as various parties are involved during the first stages of the smart card product life-cycle; details are given in threats hereafter.

Assets may overlap, in the sense that distinct assets may refer (partially or wholly) to the same piece of information or data. For example, a piece of software may be either a piece of source code (one asset) or a piece of compiled code (another asset), and may exist in various formats at different stages of its development (digital supports, printed paper). This separation is motivated by the fact that a threat may concern one form at one stage, but be meaningless for another form at another stage.

The assets to be protected by the TOE are listed below. They are grouped according to whether it is data created by and for the user (User data) or data created by and for the TOE (TSF data). For each asset it is specified the kind of dangers that weigh on it.

3.1.1.1 User data

D.APP_CODE

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

D.APP_C_DATA

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

D.APP_I_DATA


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

D.APP_KEYS

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

D.PIN

Any end-user's PIN.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

To be protected from unauthorized disclosure and modification.

3.1.1.2 TSF data

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.

To be protected from unauthorized disclosure and modification.

D.JCS_CODE

The code of the Java Card System.

To be protected from unauthorized disclosure and modification.

D.JCS_DATA

The internal runtime data areas necessary for the execution of the 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.

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.

To be protected from unauthorized disclosure and modification.

3.2 Threats

3.2.1 Java Card System Protection Profile - Open Configuration


This section introduces the threats to the assets against which specific protection within the TOE or its environment is required. Several groups of threats are distinguished according to the configuration chosen for the TOE and the means used in the attack. The classification is also inspired by the components of the TOE that are supposed to counter each threat.

3.2.1.1 CONFIDENTIALITY

T.CONFID-APPLI-DATA

The attacker executes an application to disclose data belonging to another application. See #.CONFID-APPLI-DATA [PP-JCS] for details.

Directly threatened asset(s): D.APP_C_DATA, D.PIN, and D.APP_KEYS.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

T.CONFID-JCS-CODE

The attacker executes an application to disclose the Java Card System code. See #.CONFID-JCS-CODE [PP-JCS] for details.

Directly threatened asset(s): D.JCS_CODE.

T.CONFID-JCS-DATA

The attacker executes an application to disclose data belonging to the Java Card System. See #.CONFID-JCS-DATA [PP-JCS] for details.

Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

3.2.1.2 INTEGRITY

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 [PP-JCS] for details.

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 [PP-JCS] for details.

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 [PP-JCS] for details.

Directly threatened asset(s): D.APP_I_DATA, D.PIN, 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 [PP-JCS] for details.

Directly threatened asset(s): D.APP_I_DATA and D_APP_KEYS.

T.INTEG-JCS-CODE


The attacker executes an application to alter (part of) the Java Card System code. See #.INTEG-JCS-CODE [PP-JCS] for details.

Directly threatened asset(s): D.JCS_CODE.

T.INTEG-JCS-DATA

The attacker executes an application to alter (part of) Java Card System or API data. See #.INTEG-JCS-DATA [PP-JCS] for details.

Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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 in the sequel. That is why a more detailed list is given hereafter.

3.2.1.3 IDENTITY USURPATION

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 [PP-JCS] for details.

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.

Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged).

3.2.1.4 UNAUTHORIZED EXECUTION

T.EXE-CODE.1

An applet performs an unauthorized execution of a method. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE [PP-JCS] for details.

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 [PP-JCS] for details.

Directly threatened asset(s): D.APP_CODE.

T.EXE-CODE-REMOTE

The attacker performs an unauthorized remote execution of a method from the CAD. See #.EXE-APPLI-CODE [PP-JCS] for details.

Directly threatened asset(s): D.APP_CODE.


Application note:

This threat concerns version 2.2.x of the Java Card RMI, which allow external users (that is, other than on-card applets) to trigger the execution of code belonging to an on-card applet. On the contrary, T.EXE-CODE.1 is restricted to the applets under the TSF.

T.NATIVE

An applet executes a native method to bypass a TOE Security Function such as the firewall. See #.NATIVE [PP-JCS] for details.

Directly threatened asset(s): D.JCS_DATA.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

3.2.1.5 DENIAL OF SERVICE

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 [PP-JCS] for details.

Directly threatened asset(s): D.JCS_DATA.

3.2.1.6 CARD MANAGEMENT

T.DELETION

The attacker deletes an applet or a package already in use on the card, or uses the deletion functions to pave the way for further attacks (putting the TOE in an insecure state). See #.DELETION [PP-JCS] for details).

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 [PP-JCS] for details.

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).

3.2.1.7 SERVICES

T.OBJ-DELETION

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

Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYS.


3.2.1.8 MISCELLANEOUS

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

This threatens all the identified assets.

This threat refers to the point (7) of the security aspect #.SCP, and all aspects related to confidentiality and integrity of code and data.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

3.3 Organizational Security Policies

This section describes the organizational security policies to be enforced with respect to the TOE environment.

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 [PP-JCS] for details.

3.4 Assumptions

3.4.1 Java Card System Protection Profile - Open Configuration

This section introduces the assumptions made on the environment of the TOE.

A.APPLET

Applet loaded post-issuance do not contain native methods. The Java Card specification explicitly "does not include support for native method" ([JCVM22], §3.3) outside the API.

A.DELETION

Deletion of applets through the card manager is secure. Refer to #.DELETION [PP-JCS] for details on this assumption.

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.

3.4.2 Additional assumptions

A.SID

Any applet and package are uniquely identified.


A.FIREWALL-ENV

The applet selection/deletion/loading, the object deletion, and the JCRMI ensure the controlled sharing of data.

A.GLOBAL_ARRAY_CONFID

The APDU buffer that is shared by all applications is always cleaned upon applet selection.

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.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

A.OPERATE-ENV

The applet selection/deletion/loading and the object deletion ensure the continued correct operation of the TOE security functions.

A.REALLOCATION

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.

A.RESOURCES

The availability of resources for the applications is controlled. See #.RESOURCES [PP-JCS] for details

A.ALARM-ENV

The applet deletion/loading and the object deletion ensure the appropriate feedback information upon detection of a potential security violation.

A.CIPHER

There is a means to cipher sensitive data for applications in a secure way. See #.CIPHER [PP-JCS] for details.

A.KEY-MNGT

There is a mean to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See #.KEY-MNGT.

A.PIN-MNGT

There is a mean to securely manage PIN objects. See #.PIN-MNGT [PP-JCS] for details.

A.REMOTE

Restricted remote access is provided from the CAD to the services implemented by the applets on the card. This particularly concerns the Java Card RMI services introduced in version 2.2.x of the Java Card platform.

A.TRANSACTION


There is a mean to execute a set of operations atomically. See #.TRANSACTION [PP-JCS] for details.

A.OBJ-DELETION

The object deletion shall not break references to objects. See #.OBJ-DELETION (in PP Java Card) for further details.

A.DELETION

Both applet and package deletion shall be performed as expected. See #.DELETION (in PP Java Card) for details.


	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

A.LOAD

The loading of a package into the card is safe.

A.INSTALL-ENV

The applet deletion/loading and the object deletion the installation of an applet to be performed as expected.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

4 Security Objectives

4.1 Security Objectives for the TOE

4.1.1 Java Card System Protection Profile - Open Configuration

This section defines the security objectives covered by the TOE . Note that these objectives are a subset of those ensured by the reference TOE in the PP Java Card [PPJCS], corresponding to those ensured by the components of the JCS **included** in this TOE are .

4.1.1.1 EXECUTION

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 (in [PP-JCS]) for details.

Application note:

With respect to [PP-JCS], this objective is for the components of the JCS included in this TOE, i.e. interpreter, linker and Native API. This objective is not for the following out-of-TOE components of the JCS:

- Applet selection/deletion/loading
- Object deletion
- JCRMI

that are into the TOE environment.

O.GLOBAL_ARRAYS_INTEG


The TOE shall ensure that only the currently selected application 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 (in [PP-JCS]) for details.

O.OPERATE

The card manager must ensure continued correct operation of its security functions. See #.OPERATE (in [PP-JCS]) for details.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

Application note:

With respect to [PP-JCS], this objective is for the components of the JCS included in this TOE, i.e. interpreter, linker and Native API. This objective is not for the following out-of-TOE components of the JCS:

- Applet selection/deletion/loading
- Object deletion

that are into the TOE environment.

4.1.1.2 SERVICES

O.ALARM

The TOE shall provide appropriate feedback information upon detection of a potential security violation. See #.ALARM (in [PP-JCS]) for details.

Application note:

With respect to [PP-JCS], this objective is for the components of the JCS included in this TOE, i.e. interpreter, linker and Native API. This objective is not for the following out-of-TOE components of the JCS:

- Applet selection/deletion/loading
- Object deletion

that are into the TOE environment.

4.1.1.3 Applet Management

O.INSTALL


The TOE shall ensure that the installation of an applet performs as expected. See #.INSTALL (in [PP-JCS]) for details.

Application note:

With respect to [PP-JCS], this objective is for the components of the JCS included in this TOE, i.e. interpreter, linker and Native API. This objective is not for the following out-of-TOE components of the JCS:

- Applet selection/deletion/loading
- Object deletion

that are into the TOE environment.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

4.2 Security objectives for the Operational Environment

4.2.1 Java Card System Protection Profile - Open Configuration

OE.APPLET

No applet loaded post-issuance shall contain native methods.

OE.CARD-MANAGEMENT

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

The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component will in practice be tightly connected with the TOE, which in turn shall very likely rely on the card manager for the effective enforcing of some of its security functions. Typically the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager should prevent that card content management (loading, installation, deletion) is carried out, for instance, at invalid states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.

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 [PP-JCS]for details.

OE.SCP.SUPPORT


This security objective for the environment refers to the points (2)(3)(4)(5) of the security aspect #.SCP (in [PP-JCS])

- (2) 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.
- (3) It provides secure low-level cryptographic processing to the Java Card System, GlobalPlatform.
- (4) 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.
- (5) 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).

OE.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.

This security objective for the environment 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

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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.

OE.SCP.IC

The SCP shall provide all IC security features against physical attacks.

This security objective for the environment refers to the point (7) of the security aspect #.SCP (in [PP-JCS]):

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.

4.2.2 Additional security objectives

OE.SID

The Applet selection component (part of JCS) shall uniquely identify every subject (applet, or package) before granting it access to any service.

OE.FIREWALL-ENV

The controlled sharing of data containers owned by applets of different packages, or the JCRE and between applets and the TSFs shall be ensured by the following components:

- Applet selection/deletion/loading
- Object deletion
- JCRMI

OE.GLOBAL_ARRAYS_CONFID

The Applet selection/de-selection component (part of JCS) shall ensure that the APDU buffer that is shared by all applications is always cleaned upon applet selection.

The JCS 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.


OE.OPERATE-ENV

The continued correct operation of the TOE security functions shall be ensured in the following (out-of-TOE) components:

- Applet selection/deletion/loading
- Object deletion

OE.REALLOCATION

The JCS 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.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

Application note:

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

OE.RESOURCES

The card manager/JCS shall control the availability of resources for the applications. See #.RESOURCES [PP-JCS] for details.

OE.ALARM-ENV

The appropriate feedback information upon detection of a potential security violation shall be provided in the following (out-of-TOE) components:

- Applet deletion/loading
- Object deletion

OE.CIPHER

The Cryptographic API (part of JCS) shall provide a means to cipher sensitive data for applications in a secure way. In particular, the Cryptographic API must support cryptographic algorithms consistent with cryptographic usage policies and standards. See #.CIPHER [PP-JCS] for details.

OE.KEY-MNGT

The Cryptographic API shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See #.KEY-MNGT.

Application note:

O.KEY-MNGT is actually provided to applets in the form of Java Card API. Vendor-specific libraries can also be present on the card and made available to applets; those may be built on top of the Java Card API or independently. Depending on whether they contain native code or not, these proprietary libraries will need to be evaluated together with the TOE or not (see #.NATIVE [PP-JCS]). In any case, they are not included in the Java Card System for the purpose of the present document.

OE.PIN-MNGT


The JCS shall provide a means to securely manage PIN objects. See #.PIN-MNGT [PP-JCS] for details.

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.

OE.REMOTE

The Java Card RMI component (part of JCS) shall provide restricted remote access from the CAD to the services implemented by the applets on the card. This particularly concerns the Java Card RMI services introduced in version 2.2.x of the Java Card platform.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

OE.TRANSACTION

The JCS must provide a means to execute a set of operations atomically. See #.TRANSACTION [PP-JCS] for details.

OE.OBJ-DELETION

The Object deletion component (part of JCS) shall ensure the object deletion shall not break references to objects. See #.OBJ-DELETION (in PP Java Card) for further details.

OE.DELETION

The Applet deletion component (part of JCS) shall ensure that both applet and package deletion perform as expected. See #.DELETION (in PP Java Card) for details.

OE.LOAD

The Applet loading component (part of JCS) shall ensure that the loading of a package into the card is safe.

Application note:

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

OE.INSTALL-ENV

The installation of an applet shall be performed as expected in the following (out-of-TOE) components:

- Applet deletion/loading
- Object deletion

4.3 Security Objectives Rationale


4.3.1 Threats

4.3.1.1 Java Card System Protection Profile - Open Configuration

CONFIDENTIALITY

T.CONFID-APPLI-DATA This threat is countered by the security objective for the operational environment regarding bytecode verification (OE.VERIFICATION) and 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 (OE.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.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

The objectives OE.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objectives OE.SCP.RECOVERY and OE.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 (OE.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 (OE.KEY-MNGT, OE.PIN-MNGT, OE.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 (OE.GLOBAL_ARRAYS_CONFID) security objective.

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 OE.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-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) and the assumption (A.NATIVE), so no application can be run to disclose a piece of code.

The (#.VERIFICATION) security aspect is addressed by the objective for the environment OE.VERIFICATION.

The objectives OE.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.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 (OE.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 objectives OE.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objectives OE.SCP.RECOVERY and OE.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.

INTEGRITY

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

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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) and the assumption (A.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 OE.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.LOAD This threat is countered by the security objective OE.LOAD which ensures that the loading of packages is done securely and thus preserves the integrity of packages code.

By controlling the access to card management functions such as the installation, update or deletion of applets the objective OE.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 (OE.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 objectives OE.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objectives OE.SCP.RECOVERY and OE.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 (OE.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 (OE.KEY-MNGT, OE.PIN-MNGT, OE.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 (O.GLOBAL_ARRAYS_INTEG) objective.

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 OE.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 OE.LOAD which ensures that the loading of packages is done securely and thus preserves the integrity of applications data.

By controlling the access to card management functions such as the installation, update or deletion of applets the objective OE.CARD-MANAGEMENT contributes to cover this threat.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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) and the assumption (A.NATIVE), so no application can be run to disclose or modify a piece of code. The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION.


The objectives OE.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-JCS-DATA This threat is countered by bytecode verification (OE.VERIFICATION) 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 (OE.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 objectives OE.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objectives OE.SCP.RECOVERY and OE.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.

IDENTITY USURPATION

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 (OE.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 objective (OE.GLOBAL_ARRAYS_CONFID) and (O.GLOBAL_ARRAYS_INTEG). The objective OE.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 OE.SCP.RECOVERY and OE.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.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

UNAUTHORIZED EXECUTION

T.EXE-CODE.1 Unauthorized execution of a method is prevented by the objective OE.VERIFICATION. 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.EXE-CODE-REMOTE The OE.REMOTE security objective contributes to prevent the invocation of a method that is not supposed to be accessible from outside the card.

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 (A.NATIVE). OE.APPLLET 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.

DENIAL OF SERVICE

T.RESOURCES This threat is directly countered by objectives on resource-management (OE.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.


It should be noticed that, for what relates to CPU usage, the Java Card platform is single-threaded and it is possible for an ill-formed application (either native or not) to monopolize the CPU. However, a smart card can be physically interrupted (card removal or hardware reset) and most CADs implement a timeout policy that prevent them from being blocked should a card fails to answer. That point is out of scope of this Protection Profile, though.

Finally, the objectives OE.SCP.RECOVERY and OE.SCP.SUPPORT are intended to support the O.OPERATE objective of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

CARD MANAGEMENT

T.DELETION This threat is covered by the OE.DELETION security objective which ensures that both applet and package deletion perform as expected.

The objective OE.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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 OE.LOAD which ensures that the loading of a package into the card is safe.

The objective OE.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.


SERVICES

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

MISCELLANEOUS

T.PHYSICAL This threat is countered by physical protections which rely on the underlying platform and are therefore an environmental issue.

The security objectives OE.SCP.SUPPORT and OE.SCP.IC protect sensitive assets of the platform against loss of integrity and confidentiality and especially ensure the TSFs cannot be bypassed or altered. Physical protections rely on the underlying platform and are therefore an environmental issue.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

4.3.2 Organizational Security Policies

4.3.2.1 Java Card System Protection Profile - Open Configuration

OSP.VERIFICATION This policy is upheld by the security objectives of the environment 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.

4.3.3 Assumptions

4.3.3.1 Java Card System Protection Profile - Open Configuration

A.APPLET The assumption is upheld by the environmental objective OE.APPLET which ensures that no applet loaded post-issuance shall contain native methods.

A.DELETION The assumption A.DELETION is upheld by the environmental objective OE.CARD-MANAGEMENT which controls the access to card management functions such as deletion of applets.

A.VERIFICATION This assumption is upheld by the security objective on the operational environment 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.

A.SID This assumption is upheld by OE.SID

A.FIREWALL-ENV This assumption is upheld by OE.FIREWALL-ENV

A.GLOBAL_ARRAYS_CONFID This assumption is upheld by OE.GLOBAL_ARRAYS_CONFID.

A.OPERATE-ENV This assumption is upheld by OE.OPERATE-ENV

A.REALLOCATION This assumption is upheld by OE.REALLOCATION.

A.RESOURCES This assumption is upheld by OE.RESOURCES.


A.ALARM-ENV This assumption is upheld by OE.ALARM-ENV

A.CIPHER This assumption is upheld by OE.CIPHER.

A.KEY-MNGT This assumption is upheld by OE.KEY-MNGT.

A.PIN-MNGT This assumption is upheld by OE.PIN-MNGT.

A.REMOTE This assumption is upheld by OE.REMOTE.

	Reference	D1185035	Release	1.5p (Printed copy not controlled: verify the version before using)
	Classification level	Gemalto Restricted	Pages	77

A.TRANSACTION This assumption is upheld by OE.TRANSACTION.

A.OBJ-DELETION This assumption is upheld by OE.OBJ-DELETION.


A.DELETION This assumption is upheld by OE.DELETION.

A.LOAD This assumption is upheld by OE.LOAD.

A.INSTALL-ENV This assumption is upheld by OE.INSTALL-ENV.

4.3.4 SPD and Security Objectives


Threats	Security Objectives	Rationale
T.CONFID-APPLI-DATA	OE.CARD-MANAGEMENT , OE.SID , O.OPERATE , O.FIREWALL , OE-FIREWALL-ENV , OE.REALLOCATION , OE.GLOBAL ARRAYS CONFID , O.ALARM , OE.ALARM-ENV , OE.TRANSACTION , OE.CIPHER , OE.PIN-MNGT , OE.KEY-MNGT , OE.SCP.SUPPORT , OE.SCP.RECOVERY , OE.BASIC-APPS-VALIDATION , OE.VERIFICATION	Section 4.3.1
T.CONFID-JCS-CODE	OE.CARD-MANAGEMENT , O.NATIVE , OE.VERIFICATION	Section 4.3.1
T.CONFID-JCS-DATA	OE.CARD-MANAGEMENT , OE.SID , O.OPERATE , OE.OPERATE-ENV , O.FIREWALL , OE.FIREWALL-ENV , O.ALARM , OE.ALARM-ENV , OE.SCP.SUPPORT , OE.SCP.RECOVERY , OE.VERIFICATION	Section 4.3.1
T.INTEG-APPLI-CODE	OE.CARD-MANAGEMENT , O.NATIVE , OE.VERIFICATION	Section 4.3.1
T.INTEG-APPLI-CODE.LOAD	OE.LOAD , OE.CARD-MANAGEMENT	Section 4.3.1
T.INTEG-APPLI-DATA	OE.CARD-MANAGEMENT , OE.SID , O.OPERATE , OE.OPERATE-ENV , O.FIREWALL , OE.FIREWALL-ENV , OE.REALLOCATION , O.GLOBAL ARRAYS INTEG , O.ALARM , OE.ALARM-ENV , OE.TRANSACTION , OE.CIPHER , OE.PIN-MNGT , OE.KEY-MNGT , OE.SCP.SUPPORT , OE.SCP.RECOVERY , OE.VERIFICATION , OE.BASIC-APPS-VALIDATION	Section 4.3.1
T.INTEG-APPLI-DATA.LOAD	OE.LOAD , OE.CARD-MANAGEMENT	Section 4.3.1
T.INTEG-JCS-CODE	OE.CARD-MANAGEMENT , O.NATIVE , OE.VERIFICATION	Section 4.3.1
T.INTEG-JCS-DATA	OE.CARD-MANAGEMENT , OE.SID , O.OPERATE , O.OPERATE-ENV , O.FIREWALL , OE.FIREWALL-ENV , O.ALARM , OE.ALARM-ENV , OE.SCP.SUPPORT , OE.SCP.RECOVERY , OE.VERIFICATION	Section 4.3.1
T.SID.1	OE.CARD-MANAGEMENT , O.FIREWALL , OE.FIREWALL-ENV , OE.GLOBAL ARRAYS CONFID , O.GLOBAL ARRAYS INTEG , O.INSTALL , OE.INSTALL-ENV , OE.SID	Section 4.3.1
T.SID.2	OE.SID , O.OPERATE , OE.OPERATE-ENV , O.FIREWALL ,	Section 4.3.1

	Reference	D1185035	Release	1.5p (Printed copy not controlled: verify the version before using)
	Classification level	Gemalto Restricted	Pages	77

	OE.FIREWALL-ENV, O.INSTALL , OE.INSTALL-ENV, OE.SCP.SUPPORT , OE.SCP.RECOVERY	
T.EXE-CODE.1	O.FIREWALL , OE.FIREWALL-ENV, OE.VERIFICATION	Section 4.3.1
T.EXE-CODE.2	OE.BASIC-APPS-VALIDATION , OE.VERIFICATION	Section 4.3.1
T.EXE-CODE-REMOTE	OE.REMOTE	Section 4.3.1
T.NATIVE	OE.APPLLET , O.NATIVE , OE.BASIC-APPS-VALIDATION , OE.VERIFICATION	Section 4.3.1
T.RESOURCES	O.INSTALL , OE.INSTALL-ENV, O.OPERATE , OE.OPERATE-ENV, OE.RESOURCES , OE.SCP.SUPPORT , OE.SCP.RECOVERY	Section 4.3.1
T.DELETION	OE.DELETION , OE.CARD-MANAGEMENT	Section 4.3.1
T.INSTALL	O.INSTALL , OE.INSTALL-ENV, OE.LOAD , OE.CARD-MANAGEMENT	Section 4.3.1
T.OBJ-DELETION	OE.OBJ-DELETION	Section 4.3.1
T.PHYSICAL	OE.SCP.IC , OE.SCP.SUPPORT	Section 4.3.1


Table 5 Threats and Security Objectives - Coverage

Security Objectives	Threats
O.FIREWALL	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.1 , T.SID.2 , T.EXE-CODE.1
O.GLOBAL ARRAYS INTEG	T.INTEG-APPLI-DATA , T.SID.1
O.NATIVE	T.CONFID-JCS-CODE , T.INTEG-APPLI-CODE , T.INTEG-JCS-CODE , T.NATIVE
O.OPERATE	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.2 , T.RESOURCES
O.ALARM	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA
O.INSTALL	T.SID.1 , T.SID.2 , T.RESOURCES , T.INSTALL
OE.SID	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.1 , T.SID.2
OE.FIREWALL-ENV	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.1 , T.SID.2 , T.EXE-CODE.1
OE.GLOBAL ARRAYS CONFID	T.CONFID-APPLI-DATA , T.SID.1
OE.OPERATE-ENV	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.2 , T.RESOURCES

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77


OE.REALLOCATION	T.CONFID-APPLI-DATA , T.INTEG-APPLI-DATA
OE.RESOURCES	T.RESOURCES
OE.ALARM-ENV	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA
O.INSTALL-ENV	T.SID.1 , T.SID.2 , T.RESOURCES , T.INSTALL
OE.CIPHER	T.CONFID-APPLI-DATA , T.INTEG-APPLI-DATA
OE.KEY-MNGT	T.CONFID-APPLI-DATA , T.INTEG-APPLI-DATA
OE.PIN-MNGT	T.CONFID-APPLI-DATA , T.INTEG-APPLI-DATA
OE.REMOTE	T.EXE-CODE-REMOTE
OE.TRANSACTION	T.CONFID-APPLI-DATA , T.INTEG-APPLI-DATA
OE.OBJ-DELETION	T.OBJ-DELETION
OE.DELETION	T.DELETION
OE.LOAD	T.INTEG-APPLI-CODE.LOAD , T.INTEG-APPLI-DATA.LOAD , T.INSTALL
OE.SCP.SUPPORT	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.2 , T.RESOURCES , T.PHYSICAL
OE.SCP.RECOVERY	T.CONFID-APPLI-DATA , T.CONFID-JCS-DATA , T.INTEG-APPLI-DATA , T.INTEG-JCS-DATA , T.SID.2 , T.RESOURCES
OE.SCP.IC	T.PHYSICAL
OE.APPLET	T.NATIVE
OE.CARD-MANAGEMENT	T.CONFID-APPLI-DATA , T.CONFID-JCS-CODE , T.CONFID-JCS-DATA , T.INTEG-APPLI-CODE , T.INTEG-APPLI-CODE.LOAD , T.INTEG-APPLI-DATA , T.INTEG-APPLI-DATA.LOAD , T.INTEG-JCS-CODE , T.INTEG-JCS-DATA , T.SID.1 , T.DELETION , T.INSTALL
OE.VERIFICATION	T.CONFID-APPLI-DATA , T.CONFID-JCS-CODE , T.CONFID-JCS-DATA , T.INTEG-APPLI-CODE , T.INTEG-APPLI-DATA , T.INTEG-JCS-CODE , T.INTEG-JCS-DATA , T.EXE-CODE.1 , T.EXE-CODE.2 , T.NATIVE

Table 6 Security Objectives and Threats - Coverage

	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

Organizational Security Policies	Security Objectives	Rationale
OSP.VERIFICATION	OE.VERIFICATION	Section 4.3.2

Table 7 OSPs and Security Objectives - Coverage

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

5 Security Requirements


5.1 Security Functional Requirements

5.1.1 Java Card System Protection Profile - Open Configuration

This section states the security functional requirements for the Java Card System - Open configuration. 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"), objects (prefixed with an "O") and Information (prefixed with an "I") are described in the following table:

Subject/Object/Information	Description
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.
S.JCRE	The Java Card RE is responsible on behalf of the card issuer of the bytecode execution and runtime environment functionalities. It is the process that manages applet selection and de-selection, along with the delivery of APDUs from and to the smart card device. This subject is unique
S.JCVM	The Java Card VM, is the bytecode interpreter. This subject dynamically enforces the firewall, that is, 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.APPLET	Any applet instance.
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.
O.APPLET	Any installed applet, its code and data.
I.DATA	JCVM Reference Data: objectref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77


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

Security attribute	Description/Value
Context	Package AID, or "Java Card RE"
Sharing	Standards, SIO, Java Card RE entry point, or global array
LifeTime	CLEAR_ON_DESELECT or PERSISTENT (*).
Selected Applet Context	Package AID, or "None"
Currently Active Context	Package AID, or "Java Card RE"
Package AID	The AID of each package indicated in the export file
Applet's version number	The version number of an applet (package) indicated in the export file
Dependent package AID	Allows the retrieval of the Package AID and Applet's version number ([JCV22], §4.5.2)
Registered applet AID	The AID of the applet instance registered on the card
Applet Selection Status	"Selected" or "Deselected"
LC Applet Selection Status	Multiselectable, Non-multiselectable or "None" (logical channel)
ActiveApplets	Which is a list of the active applets' AIDs
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).
Address space	Accessible memory portion

(*) 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 a specific number of parameters given between brackets, among which there is the "accessed object", the first one, when applicable. Parameters may be seen as security attributes that are under the control of the subject performing the operation.

Operation	Description
OP.ARRAY_ACCESS(O.JAVAOBJECT, field)	Read/Write an array component.
OP.INSTANCE_FIELD(O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming language

	Reference	D1185035	Release	1.5p
	Classification level	Gemalto Restricted	(Printed copy not controlled: verify the version before using)	
			Pages	77

Operation	Description
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.INVK_STATIC(method, arg1,...)	Invoke a static method.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [JCRE22],§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).
OP.JAVA(...)	Any access in the sense of [JCRE22], §6.2.8. In our formalization, this is one of the preceding operations.
OP.CREATE(Sharing, LifeTime) (*)	Creation of an object (new or makeTransient call).
OP.PUT(S1,S2,I)	Transfer a piece of information I from S1 to S2.

(*) For this operation, there is no accessed object; the "Sharing value" thus refers to the parameter of the operation. This rule simply enforces that shareable transient objects are not allowed. Note: parameters can be seen as security attributes whose value is under the control of the subject. For instance, during the creation of an object, the JavaCardClass attribute's value is chosen by the creator.

5.1.1.1 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,

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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:

	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

- 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 or OP.INVK_STATIC).

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,

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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 authorise 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.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

FMT_MSA.3/FIREWALL Static attribute initialisation

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. 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).

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 initialisation


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

FMT_MSA.3.2/JCVM 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:

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. At the creation of an

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

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).

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_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.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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 **O.JAVAOBJECTS**.

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

Application note:

FDP_ROL.1.2/FIREWALL 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). It should be noticed that the rollback within the scope of the uninstall() method only applies to Java Card platform, version 2.2.1 compliant TOEs.

Card Security Management


FAU_ARP.1 Security alarms

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

Refinement:

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

- Abortion 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
- [assignment: list of other runtime errors]

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Application note:

The developer shall provide the exhaustive list of actual potential security violations the TOE reacts to. For instance, other runtime errors related to applet's failure like uncaught exceptions.

The bytecode verification defines a large set of rules used to detect a "potential security violation". The actual monitoring of these "events" within the TOE only makes sense when the bytecode verification is performed on-card.

Depending on the context of use and the required security level, there are cases where the card manager and the TOE must work in cooperation to detect and appropriately react in case of potential security violation. This behavior must be described in this component. It shall detail the nature of the feedback information provided to the card manager (like the identity of the offending application) and the conditions under which the feedback will occur (any occurrence of the `java.lang.SecurityException` exception).

The "locking of the card session" may not appear in the policy of the card manager. Such measure should only be taken in case of severe violation detection; the same holds for the re-initialization of the Java Card System. Moreover, the locking should occur when "clean" re-initialization seems to be impossible.

The locking may be implemented at the level of the Java Card System as a denial of service (through some systematic "fatal error" message or return value) that lasts up to the next "RESET" event, without affecting other components of the card (such as the card manager). Finally, because the installation of applets is a sensitive process, security alerts in this case should also be carefully considered herein.

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 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 ([JCRE22]). 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 [JCRE22], §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 [JCV22] specification;

The API tokens defined in the export files of reference implementation

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

when interpreting the TSF data from another trusted IT product.

Application note:

FPT_TDC.1.1:

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, namely concerning memory management, I/O functions, cryptographic functions, and so on.

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.


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 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).

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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**.

5.1.1.2 InstG Security Functional Requirements

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 install a package/applet** in the following cases:

- the applet package as identified by the package AID is already resident on the card.
- the applet package contains an applet with the same Java Card name as that of another applet already resident on the card.
- the applet package references a package that is not resident on the card.

The other error cases that are mention §11.1.5 of [JCRE22] are not in the scope of the TOE.

Application note:

The TOE may provide additional feedback information to the card manager in case of potential security violations (see FAU_ARP.1).

	Reference	D1185035	Release	1.5p
			(Printed copy not controlled: verify the version before using)	
	Classification level	Gemalto Restricted	Pages	77

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 **[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 note:

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.

FPT_RCV.3.3/Installer:

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

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.

5.2 Security Assurance Requirements

For the linqUs USIM 128K product evaluation, the security assurance requirement level is EAL4 augmented with AVA_VAN.5 and ALC_DVS.2.

The *formal assurances on the Java Card virtual Machine of the linqUs USIM 128K* product augments the security assurances requirement level on the development of the TSF with:

- ADV_SPM.1
- ADV_FSP.6
- ADV_TDS.6
- ADV_IMP.2
- ADV_INT.3

5.3 Security Requirements Rationale

5.3.1 Objectives

5.3.1.1 Security Objectives for the TOE


Java Card System Protection Profile - Open Configuration

EXECUTION

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). 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_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, and FMT_MSA.1/JCVM) also indirectly contribute to meet this objective.

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 FDP_ACF.1/FIREWALL that 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

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

method. This objective mainly relies on the environmental OE.APPLET, which upholds the assumption A.APPLET.

OE.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, 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), FIA_ATD.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. But this latter SFR component is not mandatory in [JCRE22], but appears in most of security requirements documents for masked applications. Testing could also occur randomly. Otherwise, self-tests may become mandatory in order to comply to FIPS certification [FIPS 140-2].

SERVICES

O.ALARM This security objective is met by FPT_FLS.1/Installer, FPT_FLS.1 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.

APPLET MANAGEMENT

O.INSTALL This security objective specifies that installation of applets must be secure. In particular, the TSFs are protected against possible failures of the installer (FPT_FLS.1/Installer, FPT_RCV.3/Installer). Note that the security of the applet loading/deleting is not included in this objective.


	Reference	D1185035	Release	1.5p (Printed copy not controlled: verify the version before using)
	Classification level	Gemalto Restricted	Pages	77

5.3.2 Rationale tables of Security Objectives and SFRs

Security Objectives	Security Functional Requirements	Rationale
O.FIREWALL	FDP_ACC.2/FIREWALL , FDP_ACF.1/FIREWALL , FDP_IFC.1/JCVM , FDP_IFF.1/JCVM , FMT_MSA.1/JCVM , FMT_MSA.2/FIREWALL_JCVM , FMT_MSA.3/JCVM , FMT_SMF.1 , FMT_SMR.1 , FMT_MTD.1/JCRE , FMT_MTD.3/JCRE , FMT_MSA.3/FIREWALL	Section 4.3.1
O.GLOBAL_ARRAYS_INTEG	FDP_IFC.1/JCVM , FDP_IFF.1/JCVM	Section 4.3.1
O.NATIVE	FDP_ACF.1/FIREWALL	Section 4.3.1
O.OPERATE	FDP_ROL.1/FIREWALL , FAU_ARP.1 , FPT_FLS.1 , FIA_ATD.1/AID , FPT_FLS.1/Installer , FPT_RCV.3/Installer , FDP_ACC.2/FIREWALL , FDP_ACF.1/FIREWALL , FPT_TDC.1	Section 4.3.1
O.ALARM	FAU_ARP.1 , FPT_FLS.1 , FPT_FLS.1/Installer	Section 4.3.1
O.INSTALL	FPT_FLS.1/Installer , FPT_RCV.3/Installer	Section 4.3.1

Table 8 Security Objectives and SFRs - Coverage

Security Functional Requirements	Security Objectives
FDP_ACC.2/FIREWALL	O.FIREWALL , O.OPERATE
FDP_ACF.1/FIREWALL	O.FIREWALL , O.NATIVE , O.OPERATE ,
FDP_IFC.1/JCVM	O.FIREWALL , O.GLOBAL_ARRAYS_INTEG
FDP_IFF.1/JCVM	O.FIREWALL , O.GLOBAL_ARRAYS_INTEG
FMT_MSA.1/JCVM	O.FIREWALL
FMT_MSA.2/FIREWALL_JCVM	O.FIREWALL
FMT_MSA.3/FIREWALL	O.FIREWALL
FMT_MSA.3/JCVM	O.FIREWALL
FMT_SMF.1	O.FIREWALL
FMT_SMR.1	O.FIREWALL
FDP_ROL.1/FIREWALL	O.OPERATE
FAU_ARP.1	O.OPERATE O.ALARM
FPT_FLS.1	O.OPERATE , O.ALARM
FPT_TDC.1	O.OPERATE
FIA_ATD.1/AID	O.OPERATE
FMT_MTD.1/JCRE	O.FIREWALL

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Security Functional Requirements	Security Objectives
FMT_MTD.3/JCRE	O.FIREWALL
FPT_FLS.1/Installer	O.OPERATE , O.ALARM , O.INSTALL
FPT_RCV.3/Installer	O.OPERATE , O.INSTALL

Table 9 SFRs and Security Objectives

5.3.3 Dependencies

5.3.3.1 SFRs dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FPT_FLS.1/Installer	No dependencies	
FPT_RCV.3/Installer	(AGD_OPE.1)	AGD_OPE.1
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 , FMT_MSA.3/JCVM
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 , FMT_MSA.3/JCVM
FMT_MSA.1/JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL , FDP_IFC.1/JCVM , FMT_SMF.1 , 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 , FDP_IFC.1/JCVM , FMT_MSA.1/JCVM , FMT_SMR.1
FMT_MSA.3/FIREWALL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM , FMT_SMR.1
FMT_MSA.3/JCVM	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM , FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	(FIA_UID.1)	No dependency
FDP_ROL.1/FIREWALL	(FDP_ACC.1 or FDP_IFC.1)	FDP_ACC.2/FIREWALL , FDP_IFC.1/JCVM
FAU_ARP.1	(FAU_SAA.1)	
FPT_FLS.1	No dependencies	
FPT_TDC.1	No dependencies	
FIA_ATD.1/AID	No dependencies	
FMT_MTD.1/JCRE	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMF.1 , FMT_SMR.1
FMT_MTD.3/JCRE	(FMT_MTD.1)	FMT_MTD.1/JCRE


	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Table 10 SFRs dependencies

Rationale for the exclusion of dependencies


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. Potential violation analysis is used to specify the set of auditable events whose occurrence or accumulated occurrence held to indicate a potential violation of the SFRs, and any rules to be used to perform the violation analysis. The dependency of FAU_ARP.1 on this functional requirement assumes that a "potential security violation" is an audit event indicated by the FAU_SAA.1 component. The events listed in FAU_ARP.1 are, on the contrary, merely self-contained ones (arithmetic exception, ill-formed bytcodes, 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, in principle there would be no applicable audit recording in this framework. Moreover, no specification of one such recording is provided elsewhere. Therefore no set of auditable events could possibly be defined.

The dependency FMT_SMR.1 of FIA_UID.1 is unsupported because the applet selection is not in the TSF.

5.3.3.2 SARs dependencies

Requirements	CC Dependencies	Satisfied Dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.4 , ADV_TDS.3
ADV_SPM.1	ADV_FSP.4	ADV_FSP.6
ADV_FSP.6	(ADV_TDS.1)	ADV_TDS.6
ADV_IMP.2	(ADV_TDS.3) and (ALC_TAT.1) and ALC_CMC.5	ADV_TDS.3 , ALC_TAT.1 , ALC_CMC.5
ADV_TDS.6	(ADV_FSP.6)	ADV_FSP.6
ADV_INT.3	ADV_IMP.1 and ADV_TDS.3 and ALC_TAT.1	ADV_IMP.2 and ADV_TDS.6 and ALC_TAT.1
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.4
AGD_PRE.1	No dependencies	
ALC_CMC.5	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.4 , ALC_DVS.2 , ALC_LCD.1
ALC_CMS.4	No dependencies	
ALC_DEL.1	No dependencies	
ALC_DVS.2	No dependencies	
ALC_LCD.1	No dependencies	

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77


Requirements	CC Dependencies	Satisfied 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 , ASE_INT.1 , 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 , 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 , ASE_INT.1 , ASE_REQ.2
ATE COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.4 , ATE_FUN.1
ATE DPT.1	(ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1)	ADV_ARC.1 , ADV_TDS.3 , 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 , AGD_OPE.1 , AGD_PRE.1 , ATE_COV.2 , 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 , ADV_FSP.4 , ADV_IMP.1 , ADV_TDS.3 , AGD_OPE.1 , AGD_PRE.1 , ATE_DPT.1

Table 11 SARs dependencies

5.3.4 Rationale for the Security Assurance Requirements

The LinquUs 128K product is evaluated at the EAL4 level is required for this type of product since it is intended to defend against sophisticated attacks. This evaluation assurance level allows a developer to gain maximum assurance from positive security engineering based on good practices. EAL4 represents the highest practical level of assurance expected for a commercial grade product. In order to provide a meaningful level of assurance that the TOE and its embedding product provide an adequate level of defense against such attacks: the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL4.

This product is intended to be used in an open environment where sensitive and non sensitive but hostile applications will co-exist on the product. One of the most sensitive functions of the embedded software of this product, providing the property of isolation between applications is **the firewall**. To provide assurance on the correct behavior of this security function, this security target provides formal assurances on its development from the EAL7 level. The formal assurances provide evidence that this function has been implemented correctly with respect to the specification.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

Combined with EAL4+ evaluation, this evaluation will provide the highest security assurance requirements on the robustness and on the correctness of such a security function.


5.3.4.1 ADV_SPM.1 Formal TOE security policy model

The formally modelled security policies consist of:

1. Firewall security policy that controls the sharing of data containers owned by applets of different packages, or the JCRE and between applets and the TSFs (cf. **O.FIREWALL** objective)
2. Typing security policy that supposes that all the byte-codes are verified before being loaded/installed/executed on the TSF (cf. **A.VERIFICATION** assumption)
3. (Static) information access security policy that ensures that an external reference is accessible in the current package only if this reference is exported by its package and that package is imported by the current package (part of **O.INSTALL** objective)

The SPM component is fulfilled by the following evident elements.

Requirement	Title	Type	Reference
TOE Security Policy	Interpreter	Document	D1187334_1
	linker (static access control)	Document	D1187334_2
	BCV/Typing	Document	D1187334_3
ADV_SPM.1.1C	SPM formal model	Coq code	D1187336
	Model of the JC virtual machine	Document	D1187337_1
	Model of the Firewall policy	Document	D1187337_2
	Model of the Typing policy	Document	D1187337_3
	Correspondence between ISP and TSP model	Document	D1187343
ADV_SPM.1.2C	FWVM state machine and JCVM/FWVM proof	Coq code	D1187341
	Proof of OT.Firewall: confidentiality	Document	D1187334_4
	TYVM state machine and JCVM/TYVM proof	Coq code	D1187342
	Proof of the safe execution of a CAP file	Document	D1187334_5
	Proof of OT.GLOBAL_ARRAYS_INTEG	Coq code	
	Proof of O.FIREWALL: integrity	Coq code	
ADV_SPM.1.3C ADV_SPM.1.4C ADV_SPM.1.5C	Correspondence proof between the FSP and the TSP formal models (linker, interpreter, API)	Coq code	D1187350
	JC22 Linker: Correspondence between FSP and TSP models	Document	D1187351
	JC22 interpreter: Correspondence between FSP and TSP models	Document	D1187352

	Reference	D1185035	Release	1.5p
	Classification level	Gemalto Restricted	(Printed copy not controlled: verify the version before using)	
			Pages	77

	JC22 API: Correspondence between the FSP and the TSP formal models	Document	D1187348
--	--	----------	----------

It depends on ADV_FSP.4 that is satisfied by this evaluation.

5.3.4.2 ADV_FSP.6 Complete semi-formal functional specification with additional formal specification

This requirement is fulfilled by the following evident elements.


Requirement	Title	Type	Reference
ADV_FSP.6	JC22 Linker: FSP model	Coq code	D1187345
	JC22 Interpreter: FSP model		
	JC22 API (native): FSP model		
	JC22VM: Functional Spec of the Linker	Document	D1187347_1
	JC22VM: Functional Spec of the Interpreter	Document	D1187347_2
	JC22 native API: formal FSP, HLD and LLD models and proofs	Document	D1187348
	Correspondence between ST (SFRs) and FSP model	Document	D1187349

It depends on ADV_TDS.1 that is satisfied by this evaluation.

5.3.4.3 ADV_TDS.6 Complete semi-formal modular design with formal high-level design presentation

This requirement is fulfilled by the following evident elements.

Requirement	Title	Type	Reference
ADV_TDS.6 (subsystem)	JC22 Linker: HLD model	Coq code	D1187354
	JC22 embedded interpreter: HLD model		
	JC22 API: HLD model		
	JC22 Linker: High-Level Design	Document	D1187359_1
	JC22 Interpreter: High-Level Design	Document	D1187359_2
	JC22 API: High-Level Design	Document	D1187348
ADV_TDS.6 (proofs for subsystems)	Correspondence between the HLD and the FSP formal models (linker, interpreter, API)	Coq code	D118736
	JC22 Linker: FSP-HLD correspondence	Document	D1187362
	JC22 interpreter: FSP-HLD correspondence	Document	D1187363
ADV_TDS.6 (modules)	JC22 interpreter and API: LLD model	Coq code	D1187368
	JC22 Linker: Low-Level Design	Document	D1187369
	JC22 Interpreter: Low-Level Design	Document	D1187371

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

ADV_TDS.6 (proofs for module)	Correspondence between the LLD and the HLD formal models (interpreter, API)	Coq code	D1187373
	Correspondence between the LLD and the HLD formal models (interpreter)	Document	D1187374

It depends on ADV_FSP.6 that is satisfied by this evaluation.

5.3.4.4 ADV_IMP.2 Implementation of the TSF

This requirement is fulfilled by the following evident elements.

Title	Type
TSF source code (in HTML format)	HTML

It depends on ADV_TDS.3, ALC_TAT.1, ALC_CMC.5 that are satisfied by this evaluation

5.3.4.5 ADV_INT.3 Minimally complex internals

This requirement is fulfilled by the following evident elements.

Requirement	Title	Type	Reference
ADV_INT.3	Minimally complex internals	Document	D1187377

It depends on ADV_IMP.1, ADV_TDS.3 and ALC_TAT.1 that are all satisfied by this evaluation.

5.3.4.6 ALC_CMC.5 Advanced support

This requirement is fulfilled by [CMC_PLF].

It depends on ALC_CMS.1, ALC_DVS.2 and ALC_LCD.1 that are all satisfied by this evaluation.


5.3.4.7 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE is intended to operate in hostile environments. AVA_VAN.5 "Advanced methodical vulnerability analysis" is considered as the expected level for Java Card technology-based products hosting sensitive applications. AVA_VAN.5 has dependencies on ADV_ARC.1, ADV_FSP.1, ADV_TDS.3, ADV_IMP.1, AGD_PRE.1, AGD_OPE.1. All of them are satisfied by EAL4.

5.3.4.8 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE and the embedding product. The standard ALC_DVS.1 requirement mandated by EAL4 is not enough. Due to the nature of the TOE and embedding product, it is necessary to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

6 TOE Summary Specification

6.1 TOE Summary Specification

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 [JCAPI].

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 realizes applet bytecode execution according to JVM rules [JVM].

The JVM execution may be summarized in JVM interpreter start-up, bytecode execution and JVM 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 JVM manages 5 types of objects: persistent objects, transient objects, persistent arrays (Boolean, byte, short, integer or reference), transient arrays (Boolean, byte, short, integer or reference) and static field images. For each type of object, different types of control are performed [see JVM §4].

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
- 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.

JCS.Firewall

This security function enforces the Firewall access control policy and the JVM 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.

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77

The JCRE shall allocate and manage a context for each Java package containing applets. The JCRE maintains its own context a special system privileges so that it can perform operations that are denied to contexts of applets.

JCS.Package

This security functions manages packages. Package is a structural item defined for naming, linking, and execution context definition. There are rules for identification of package, for structure check and access rules definition. If inconsistent items are found in a package, an error message is sent.

JCS.RunTimeExecution

This security function provides a secure run time environment and deals with:

- Bytecode execution
- JCS API methods execution
- Logical channel management
- JCRE memory and context management
- JCRE access rights
- JCRE throw exception
- JCRE security reaction

JCS.Transaction

The security function performs write operations atomically on persistent memory in order to avoid incomplete update. Prior to be written, the data are stored in a back-up area. In case of writing interrupt, the only two possible values are: initial value if writing is not started or final value if writing is started.

JCS.Install

This security function performs the installation of loaded executable files.


6.2 SFRs and TSS

6.2.1 SFRs and TSS - Rationale

Chapter content has been removed in Public version.


6.2.2 Association tables of SFRs and TSS

Chapter content has been removed in Public version.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

7 Notice


This document has been generated with TL SET version 2.3.7 (for CC3). For more information about the security editor tool of Trusted Labs visit our website at www.trusted-labs.com.

	Reference	D1185035	Release	1.5p (Printed copy not controlled: verify the version before using)
	Classification level	Gemalto Restricted	Pages	77

8 References, Glossary and Abbreviations


8.1 External References

Reference	Title
[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCIMB-2009-07-001, version 3.1 Release 3, July 2009.
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements CCIMB-2009-07-002, version 3.1 Release 3, July 2009.
[CC-3]	Common Criteria for Information Technology security Evaluation Part 3: Security Assurance Requirements CCIMB-2009-07-003, version 3.1 Release 3, July 2009.
[CEM]	Common Methodology for Information Technology Security Evaluation CCIMB-2009-07-004, version 3.1 Release 3, July 2009.
[Note/12.1]	Note d'Application - Modélisation formelle des politiques de sécurité d'une cible d'évaluation 587/SGDN/DCSSI/SDR Référence : NOTE/12.1
[AIS34]	Evaluation Methodology for CC Assurance Classes for EAL5+ ; version 1.0 ; June 2004; Ref. AIS 34
[Comp]	CCDB, Composite product evaluation for Smart Cards and similar devices, September 2007, Version 1.0 - Revision 1, September 2007, CCDB-2007-09-001
[GP22]	Global Platform 2.2.2, Specification GP
[GP-CCCM]	GlobalPlatform, Card Confidential Card Content Management, Card specification v2.2 – Amendment A,
[ISO 7816-4]	Identification cards - Integrated circuit(s) cards with contacts, Part 4: Interindustry commands for interchange.
[ISO 7816-6]	Identification cards - Integrated circuit(s) cards with contacts, Part 6: Interindustry data elements.
[ISO 7816-9]	Identification cards - Integrated circuit(s) cards with contacts, Part 9: Additional Inter industry commands and security attributes.
[ISO 9796-2]	ISO/IEC 9796-2 Information technology -- Security techniques -- Digital signature schemes giving message recovery -- Part 2: Integer factorization based mechanisms
[JCAPI222]	Java Card™ APIs specification version 2.2.2, Sun Microsystems, Inc, March 2006.
[JCRE222]	Java Card™ Runtime Environment Specification version 2.2.2, Sun Microsystems, Inc, March 2006
[JVM222]	Java Card 2.2.2 Virtual Machine Specification, Sun Microsystems, March 2006
[JIL]	Joint Interpretation Library Composite product evaluation for Smart Cards and similar devices Version 1.0 September 2007
[PP-BSI-0035]	Security IC Platform Protection Profile Version 1.0 15.06.2007
[PP-JCS]	Java Card™ System Protection Profile "Open Configuration" Version 2.6
[PP-USIM]	(U)SIM Java Card Platform Protection Profile Basic Configuration V2.0.2, June 2010
[ST/IC]	Security Target of SC33F640 SMD_SC33F640_ST_10_001 Rev 01.00

	Reference D1185035	Release 1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level Gemalto Restricted	Pages 77


8.2 Internal References

Reference	Title
[ST_LinqUs128]	Security target of LinqUs USIM 128k certified using SC33F640 ST_ D1172363 Version 1.4
[IMP_PLF]	PHENIX Platform Implementation representation D1145955 (IMP_PLF_D1145955)
[CMC_PLF]	PHENIX Platform Configuration Management Plan D1145963 (CMC_PLF_D1145963)
[CMS_PLF]	PHENIX Platform Configuration Management Scope D1145965 (CMS_PLF_D1145965)

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

8.3 ABBREVIATIONS

Abbreviation	Description
AID	Applet Identifier
APDU	Application Protocol Data Unit
API	Application Programmer Interface
CC	Common Criteria
CM	Card Manager
CPLC	Card Production Life Cycle
DAP	Data Authentication Pattern
DS	Dedicated Software
EAL	Evaluation Assurance Level
GP	Global Platform
IC	Integrated Circuit
JCRE	Java Card Runtime Environment
JCS	Java Card System
JCVM	Java Card Virtual Machine
MAC	Message Authentication Code
OSP	Organizational Security Policy
PP	Protection Profile
ST	Security Target
TOE	Target of Evaluation.

	Reference	D1185035	Release	1.5p <small>(Printed copy not controlled: verify the version before using)</small>
	Classification level	Gemalto Restricted	Pages	77

8.4 Glossary

Term	Definition
Application	Instance of an Executable Module after it has been installed and made selectable
APDU	Standard communication messaging protocol between a card accepting device and a smart card
DAP Block	Part of the Load File used for ensuring Load File Data Block verification
DAP Verification	A mechanism used by a Security Domain to verify that a Load File Data Block is authentic
Delegated Management	Pre-authorized Card Content changes performed by an approved Application Provider
Executable Load File	Actual on-card container of one or more application's executable code (Executable Modules). It may reside in Immutable Persistent Memory or may be created in Mutable Persistent Memory as the resulting image of a Load File Data Block.
Executable Module	Contains the on-card executable code of a single application present within an Executable Load File
Issuer Security Domain	The primary on-card entity providing support for the control, security, and communication requirements of the card administrator (typically the Card Issuer or MNO)
Load File	A file transferred to a GlobalPlatform card that contains a Load File Data Block and possibly one or more DAP Blocks
Load File Data Block	Part of the Load File that contains one or more application(s) or libraries and support information for the application(s) as required by the specific platform
Load File Data Block Hash	A value providing integrity for the Load File Data Block
Message Authentication Code (MAC)	A symmetric cryptographic transformation of data that provides data origin authentication and data integrity
Secure Channel	A communication mechanism between an off-card entity and a card that provides a level of assurance, to one or both entities
Secure Channel Protocol	A secure communication protocol and set of security services
Security Domain	On-card entity providing support for the control, security, and communication requirements of an off-card entity (e.g. the Card Issuer, an Application Provider or a Controlling Authority)
Supplementary Security Domain	A Security Domain other than the Issuer Security Domain
Token	A cryptographic value provided by a Card Issuer as proof that a Delegated Management operation has been authorized

- END OF DOCUMENT -