

Security Target

IDMotion V2 Virtual Machine

CONTENT

1. REFERENCES.....	6
1.1 EXTERNAL REFERENCES	6
2. INTRODUCTION.....	7
2.1 SECURITY TARGET REFERENCE	7
2.2 TOE REFERENCE	7
2.3 SECURITY TARGET IDENTIFICATION	7
2.4 TOE REFERENCE	8
2.5 SECURITY TARGET OVERVIEW.....	8
2.6 TARGET OF EVALUATION DESCRIPTION	9
2.6.1 Product Description.....	9
2.6.2 TOE functions	11
2.6.3 Intended Method of Use	11
2.6.4 Actors.....	13
2.6.5 Smartcard Product Life Cycle.....	14
2.6.6 Target of Evaluation Location and Usage	16
2.6.7 Supporting Firmware.....	17
2.6.8 Supporting Security Infrastructure	17
2.6.9 Application Load Units (ALU).....	19
2.6.10 Key Transformation Unit (KTU).....	19
2.6.11 Application Load and Delete Certificates (ALCs & ADCs)	19
2.6.12 Keys.....	20
2.6.13 MULTOS Initialization Security Data (Card Pre-enablement)	21
2.6.14 MSM Controls Data (Card enablement).....	23
2.6.15 Loading Applications (Personalisation)	24
3. CONFORMANCE CLAIMS	26
3.1 COMMON CRITERIA CONFORMANCE CLAIMS.....	26
3.2 PROTECTION PROFILE CLAIM AND PACKAGE CLAIM	26
4. SECURITY PROBLEM DEFINITION.....	27
4.1 ASSETS	27
4.2 THREATS.....	27
4.2.1 Unauthorized Full or Partial Cloning of the Target of Evaluation	28
4.2.2 Threats on Phase 1.....	28
4.2.3 Threats on Delivery for/from Phase 1 to Phases 4 to 6	29
4.2.4 Threats on Phases 4 to 7	29
4.2.5 Threats on Phases 6 to 7	30
4.2.6 Threats on Phase 7.....	31
4.3 ORGANIZATIONAL SECURITY POLICIES	32
4.4 ASSUMPTIONS	33
4.4.1 Assumptions on the Target of Evaluation Delivery Process (Phases 4 to 7)	33
4.4.2 Assumptions on Phases 4 to 6	33
4.4.3 Assumption on Phase 7	33
4.4.4 Assumption on Loaded-Application Development (Phase A1)	33
4.4.5 Additional assumptions	33
4.5 COMPOSITION TASKS	34
4.5.1 Statement of Compatibility – Threats.....	34
4.5.2 Statement of Compatibility – OSPs.....	36
4.5.3 Statement of Compatibility – Assumptions.....	37
4.5.4 Statement of Compatibility – Security Objectives	38
4.5.5 Statement of Compatibility – SFRs	40
5. SECURITY OBJECTIVES.....	42
5.1 SECURITY OBJECTIVES FOR THE TARGET OF EVALUATION	42
5.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT.....	42

IDMotion V2 Virtual Machine Security Target

5.2.1	Objectives on Phase 1	42
5.2.2	Objectives on the Target of Evaluation Delivery Process (Phases 4 to 7).....	43
5.2.3	Objectives on Delivery from Phase 1 to Phases 4, 5 and 6.....	43
5.2.4	Objectives on Phases 4 to 6	43
5.2.5	Objectives on Phase 7	43
5.2.6	Additional objectives for the operational environment	44
5.2.7	Objectives on Loaded-Application Development and Loading (Phases A1 and A2).....	45
5.3	SECURITY OBJECTIVES RATIONALE	45
5.3.1	Discussion of Threats, OSP and Security Objectives.....	45
5.3.2	Threats & OSP Addressed by Security Objectives	48
5.3.2.1	Security objectives for the TOE.....	48
5.3.2.2	Security objectives for the environment.....	50
5.3.3	Assumptions and Security Objectives for the Environment.....	57
5.3.4	Additional Assumptions and Security Objectives for the Environment.....	58
6.	EXTENDED COMPONENTS DEFINITION	60
7.	SECURITY REQUIREMENTS	61
7.1	S SUPPORTING SECURITY INFRASTRUCTURE.....	61
7.2	SECURITY FUNCTIONAL REQUIREMENTS (SFRs)	62
7.2.1	Security Audit Automatic Response (FAU_ARP).....	62
7.2.1.1	FAU_ARP.1 Security alarms.....	62
7.2.2	Security audit analysis (FAU_SAA).....	62
7.2.2.1	Potential violation analysis	62
7.2.3	Access control policy FDP_ACC.....	62
7.2.3.1	FDP_ACC.2 Complete access control	62
7.2.4	Access control functions FDP_ACF	62
7.2.4.1	FDP_ACF.1 Security attribute based access control.....	62
7.2.5	Rollback (FDP_ROL).....	63
7.2.5.1	FDP_ROL.1 Basic rollback	63
7.2.6	Fail secure (FPT_FLS).....	63
7.2.6.1	FPT_FLS.1 Failure with preservation of secure state	63
7.2.7	Trusted recovery (FPT_RCV)	64
7.2.7.1	FPT_RCV.4 Function recovery	64
7.2.8	Resource allocation (FRU_RSA)	64
7.2.8.1	FRU_RSA.1 Maximum quotas	64
7.3	SECURITY ASSURANCE REQUIREMENTS (SARs).....	65
7.4	SECURITY FUNCTIONAL REQUIREMENTS DEPENDENCIES.....	65
7.5	SECURITY REQUIREMENTS RATIONALE.....	65
7.5.1	Security Functional Requirements Rationale.....	65
7.5.1.1	SFRs Tracing Rationale	65
7.5.1.2	SFRs Justifications Rationale	66
7.5.2	SARs and the Security Requirements Rationale	67
7.5.2.1	ADV_SPM.1 Formal TOE security policy model	67
7.5.2.2	ADV_FSP.6 Complete semi-formal functional specification with additional formal specification.....	67
7.5.2.3	ADV_TDS.6 Complete semi-formal modular design with formal high-level design presentation	67
7.5.2.4	ADV_IMP.2 Implementation of the TSF.....	68
7.5.2.5	ADV_INT.3 Minimally complex internals	68
7.5.2.6	ATE_DPT.4. Testing: implementation representatio	68
7.5.2.7	ATE_COV.3. Rigorous analysis of coverage.....	68
7.5.2.8	ATE_FUN.2. Ordered Functional Testing.....	68
7.5.2.9	ALC_CMC.5 Advanced support.....	68
7.5.2.10	ALC_LCD.2 Measurable life-cycle model	68
7.5.2.11	ALC_TAT.3 Compliance with implementation standards – all parts	69
7.5.2.12	ALC_DVS.2 Sufficiency of security measures.....	69
7.5.2.13	AVA_VAN.5 Advanced methodical vulnerability analysis	69
8.	TARGET OF EVALUATION SUMMARY SPECIFICATION.....	70
8.1	SECURITY FUNCTIONALITY	70
8.1.1	Application Load Control SF (SF.Load).....	70
8.1.2	Application Delete Control SF (SF.Delete)	70
8.1.3	Application Execution Management SF (SF.Firewall)	70

IDMotion V2 Virtual Machine Security Target

8.1.4	<i>Start-up Initialization SF (SF.Init)</i>	71
8.1.5	<i>Mapping between Security Functions and Security Functional Requirements</i>	72
8.1.6	<i>TOE Security Functions Rationale</i>	72
VOCABULARY		75

FIGURES

Figure 1: Layered Structure of MULTOS Software	10
Figure 2: Smartcard IC with Multi-Application Platform Life Cycle	14
Figure 3: MULTOS Platform Life Cycle.....	16
Figure 4: MULTOS Infrastructure Context Diagram	17
Figure 5: MULTOS Initialization Security Data Information Flow	22
Figure 6: MSM Controls Data Information Flow	23
Figure 7 : Principal Key and Data Exchanges in Loading MCD Applications.....	24

TABLES

Table 1: Identification of the actors.....	13
Table 2: Product and TOE life-cycle phases.....	15
Table 3: MULTOS Security Infrastructure Keys	20
Table 4: MISA security Data.....	22
Table 5: Relationship between phases and threats.....	32
Table 6: Mapping of security objectives to threats and OSP relative to phases 4 to 7	48
Table 7: Mapping of security objectives to threats & OSP relative to phases 6 and 7.....	49
Table 8: Mapping of security objectives for the environment to threats, assumptions and OSP relative to phases 4 to 7.....	50
Table 9 : Mapping of security objectives for the environment to threats, assumptions and OSP relative to phase 1	52
Table 10 : Mapping of security objectives for the environment to threats, assumptions and OSP relative on delivery from phase 1 to phases 4 to 6.....	53
Table 11: Mapping of security objectives for the environment to threats, assumptions and OSPs on phases A1 and A2 (development and delivery for phase A1 to phases 6 and 7).....	55
Table 12: demonstrates mapping of security objectives for the operational environment to assumptions.....	57
Table 13: Mapping of additional security objectives for the operational environment to assumptions	58
Table 14: Functional dependencies in Multi-Application environment	65
Table 15: Mapping of security functional requirements and objectives	66
Table 16: Mapping between security functions and security functional requirements	72

1. REFERENCES

1.1 EXTERNAL REFERENCES

[ISO]	ISO references
[ISO7816]	Identification cards – Integrated circuit(s) cards with contacts - Books 1 to 9
[ISO14443]	Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Books 1 to 4
[CC]	Common Criteria references
[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, CCMB-2006-09-001, version 3.1 rev 5, April 2017
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, CCMB-2007-09-002, version 3.1 rev 5, April 2017
[CC-3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, CCMB-2007-09-003, version 3.1 rev 5, April 2017
[CEM]	Common Methodology for Information Technology Security Evaluation Methodology CCMB-2007-09-004, version 3.1 rev 5, April 2017
[ST-IC]	Security Target Common Criteria H13 IFX_CCI_000003h, IFX_CCI_000005h, IFX_CCI_000008h, IFX_CCI_00000Ch, IFX_CCI_000013h, IFX_CCI_000014h, IFX_CCI_000015h, IFX_CCI_00001Ch, , IFX_CCI_00001Dh Revision: 0.5 – 2017-05-22
[CR-IC]	<i>Certification Report</i> , BSI-DSZ-CC-0945-2017
[ST_full_IDMotionV2]	ID Motion V2 Platform Security Target (public version). Ref. ST_D1172991_P, Version 1.2.
[AGD]	Guidance
[MULTOS_ENABLE MENT]	MULTOS Enablement Reference: MAO-DOC-TEC-101 v1.2
[MULTOS_GLDA]	MULTOS Guide to Loading and Deleting Reference: MAO-DOC-TEC-008 v2.28
[MULTOS_MDRM]	MULTOS Developer's Reference Manual Reference: MAO-DOC-TEC-006 v1.54
[MULTOS_SGAD]	MULTOS Security Guidance for MULTOS Application Developers Reference: MI-MA-0031 v1.6
[MULTOS_GALU]	MULTOS Guide to Generating Application Load Units Reference: MAO-DOC-TEC-009 v2.9
[MULTOS_MVP]	MULTOS Mask Verification Procedure Reference: MI-PR-0012 v1.1

2. INTRODUCTION

2.1 SECURITY TARGET REFERENCE

Title	IDMotion V2 Virtual Machine Security Target
Version	1.0
ST Reference	ST_D1430933_P
Origin	Gemalto and Trusted Labs
ITSEF	Thales
Certification scheme	Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)

Common Criteria: Version 3.1, Revision 5, April 2017

Identity of the Target of Evaluation (TOE): The Target of Evaluation is IDMotion V2 Virtual Machine with OS MULTOS V4.5.2 platform mask on IFX_CCI_000014 family component (Infineon).

2.2 TOE REFERENCE

Product name	IDMotion V2
TOE name	IDMotionV2 Pultos Virtual Machine
TOE technical product reference	T1036817
Security Controllers	IFX_CCI_000014 (Infineon)
AGD documentation	[MULTOS_ENABLEMENT], [MULTOS_GLDA], [MULTOS_MDRM], [MULTOS_SGAD], [MULTOS_GALU], [MULTOS_MVP]

2.3 SECURITY TARGET IDENTIFICATION

Version of the TOE:

Each mask reference is identified by having an:

- OS type / OS version: 000452
Store in the NVM (constant defined in the source code)
- Build number: To be updated at the VLR
Stored in NVM. The build number is generated automatically by our build server and inserted into buildnumber.h before automatically building the source. This build number is used as a label in the Configuration Management System.
- Chip identity data (only byte 1 of the Chip Identification Data corresponding to the Platform Identifier): 0x16
- AMD Version Information : 01510001
Held in NVM. AMD version for IDMotionV2 is 0151v001 and has been allocated by MAOSCO and is a constant defined in the source code.

Get Configuration Data APDU Command

CLA	INS	P1	P2	Lc	Data	Le
80	10	00	00	-	-	00

Get Configuration Data APDU Response

Token	Request	Data Returned
0x00 00	Platform Identification	OS_type + OS_version
0x01 00	Largest ALU Possible	max_alu_size
0x02 00	Communication Transfer Parameters	comms_tx_parameters

IDMotion V2 Virtual Machine Security Target

0x03 00	ATR Control	cold_reset_application_id + warm_reset_application_id
0x04 00	AMD Version Information	amd_version_data
0x05 00	Codelets available	codelet_list
0x06 00	Applications loaded	{application_id + application_memory_allocated}
0x07 00	MKD_PKC	MULTOS_pk_certificate
0x08 00	Codelet checksums	The 4 byte MULTOS checksum for each codelet listed in the same orders as the codelets in token 0x0500
0x09 00	ATS Control	application_ATS_selected.application_id or MULTOS_aid
0x0A 00	Build Number	The build number of the implementation. Encoding defined by the implementer.
0x0B 00	Primitives Supported	A list of bits, one bit per possible primitive (4 sets of 256 primitives, with a "1" indicating that the primitive is supported. The first byte contains the bits for set 0 primitives 0-7 (held in bits 0-7), the second byte for set 0 primitives 8-17 and so on.
0x0C 00	Chip Identity Data	Silicon manufacturer specific chip identity data.

2.4 TOE REFERENCE

Product Name :	IDMotion V2
TOE Name :	IDMotionV2 Multos Platform
TOE Technical Product Reference*:	T1036817
Security Controllers:	IFX_CCI_000014 (Infineon)

* Note: PDM reference

The Product and TOE identification details are provided in §Product description.

2.5 SECURITY TARGET OVERVIEW

The integrated circuit card (ICC), or smartcard (or inlays/booklets, modules, or chips), is an ideal tool for the delivery of distributed, secure information processing at low cost. However, an application developed for one smartcard is usually not portable to another. Furthermore, many current smartcard operating systems allow only one application per card, meaning end users must carry a multitude of cards, one for each function or service required. Multos International, in its role as a member of the MULTOS Consortium (also known as MAOSCO), is developing an open, high-security multi-application operating system to address the current shortcomings of smartcard operating systems. This operating system is called MULTOS.

In order to satisfy the objectives set for it, MULTOS should be able to:

- Execute an application written for MULTOS - application execution should be independent of the underlying smartcard hardware.
- Load many applications - applications should be able to co-exist on the smartcard.
- Ensure that applications are securely loaded and segregated - they should not be able to interfere with each other or with MULTOS.

In summary, MULTOS provides a common development and operating platform for smartcard applications. It allows multiple applications to be loaded onto a single smartcard and execute without interfering with or being interfered with by other applications. It also allows applications written for MULTOS to execute on different types of smartcard independent of the underlying smartcard hardware.

This security target focuses on the MULTOS virtual machine that ensures the **secure execution** and the **segregation** of the applications (during **loading/deleting time** and **runtime**).

2.6 TARGET OF EVALUATION DESCRIPTION

This part of the Security Target describes the Target of Evaluation as an aid to the understanding of its security requirements and addresses the product type, the intended usage and the general IT features of the TOE.

The target of evaluation - the MULTOS Virtual Machine made of the Abstract Application Machine and the Application Memory Manager, is part of MULTOS OS (see Figure 1), embedded on the product identified in section 1.1.

2.6.1 Product Description

MULTOS is an operating system for integrated circuit cards (also known as smartcards). It is designed to allow multiple smartcard applications to be securely loaded and executed on a smartcard.

The TOE is the Virtual Machine that is composed of the AM and the MM subsystems.

- The VM subsystem ensures the interpretation of the MULTOS applications (in runtime)
- The MM subsystem ensures the management of the memory space dedicated to each application. In particular, the MM allocates this space in loading time and controls the access to its elements in runtime.

Beside the TOE, the product also contains native applications and native modules used by ICAO application (out of scope of the TOE).

Mel applications and native modules

- Native/MEL application: GMF v1.0
The GMF application was initially developed to provide a global privacy protocol (PACE) protecting the access to all applications embedded on the ID Motion product.
The GMF (Global Master File) represents – as its name says – the card's Master File and its sub-structures, files and keys. From a terminal point of view, it is an application that processes all commands performed under the Master File:
 - Selection/read of transparent EF (e.g EF.CardAccess for PACE)
 - Global Authentication commands for PACE
 - Selection of an application and re-routing of the commands to the appropriate selected application (this is transparent to the terminal).
- MEL application: Pin Server Application (PSA) v0.2
Unlike Global Platform based smartcards, MULTOS does not have a “global PIN” or Cardholder Verification Method (CVM). In the JavaCard/Global Platform world, the global CVM allows several applications to share a common, global Personal Identification Number (later PIN). This means that in a multi-applicative context, if the CVM has been verified by an application (e.g. to unlock an access condition), it does not need to be verified by all other applications that rely on the CVM validation.
As this feature is required by some of the ID Motion applications, namely IAS Classic, the service is implemented by the PSA application. In addition to managing the “standard” CVM, the PSA is in charge also of managing biometric CVMs.
- Native application: ETravel EAC/PACE/BAC v2.4
Native ICAO passport application embedded as a part of an ID Motion platform, based on MULTOS technology.
- MEL application: IAS Classic v4.4.1C
IAS is an application that provides all the necessary functions to integrate a smart card in a public key infrastructure (PKI) system, suitable for identity and corporate security applications. It is also useful for storing information about the cardholder and any sensitive data. IAS implements state-of-the-art security and conforms to the latest standards for smart cards and PKI applications. It is also fully compliant with digital signature law.
- MEL application: MOC client (MOCC) v1.0.2A
- MOC Client relies on PSA (PIN Server Application). The biometric templates are all stored within PSA and services are provided through MULTOS delegation.

IDMotion V2 Virtual Machine Security Target

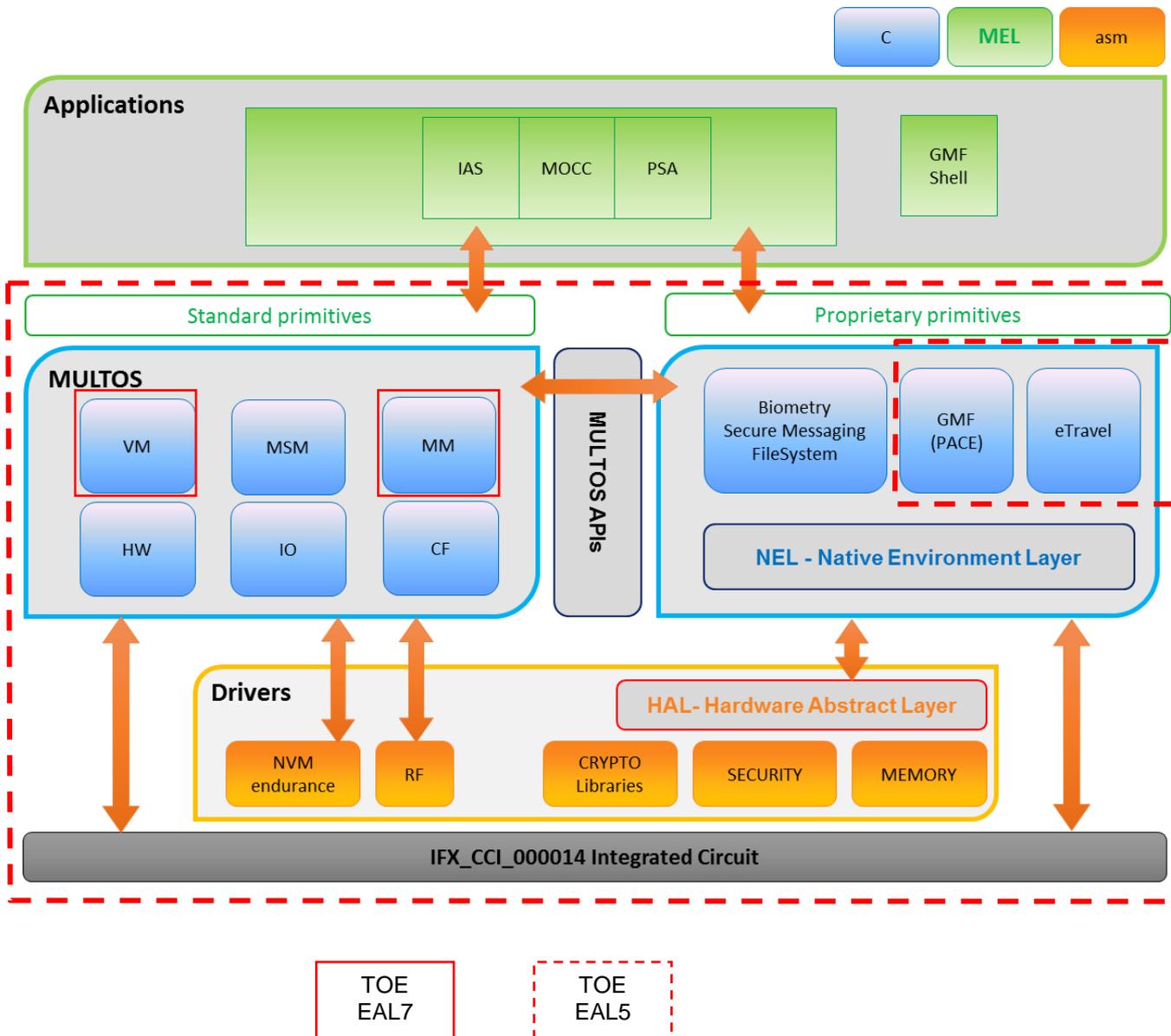


Figure 1: Layered Structure of MULTOS Software

VM: Virtual Machine

MSM: Multos Security Manager

MM: Application Memory Manager Subsystem

CF: Cryptographic Functions subsystem

IO: I/O Communications subsystem

HW: Hardware Services subsystem

NVM: Non Volatile Memory

The Biometry Secure Messaging regroups 4 native blocks:

- **MBIO:** Fingerprint Match on Card algorithm
- **MSM:** ISO Secure Messaging services (Set & Clear session, Wrap & Unwrap)
- **MFS:** This module provides each ID Motion application with the ability to construct and manipulate its own file system in the NVM static area allocated to it by MULTOS.
- **GMF:** Native Commands (MGAM) – The GMF (Global Master File) sub-system represents – as its name says – the card's Master File and its sub-structures, files and keys. From a terminal point of view, it is an application that processes all commands performed under the Master File:
 - o Selection/read of transparent EF (e.g EF.CardAccess for PACE)

Global Authentication commands (e.g PACE authentication).

IDMotion V2 Virtual Machine Security Target

The user of the smartcard accesses the applications loaded on the MULTOS operating system via an Interface Device (IFD), which could be a Point-of-Sale terminal, Automatic Teller Machine, or some other device which supports ISO 7816 smartcard protocols.

Communications across the IFD-MULTOS interface comprise a message transmitted by the smartcard when it is reset (the Answer-to-Reset or ATR message), followed by command-response pairs, where a command is a message from the IFD to MULTOS and a response is a message from MULTOS to the IFD.

By means of these command-response pairs, MULTOS allows:

- a) Applications to be loaded into and deleted from the smartcard.
- b) An IFD to access data and applications which are loaded on the card.
- c) Information specific to the card to be retrieved by an IFD.

MULTOS is a single-threaded operating system. Only one application can be executing at any given time. MULTOS does not provide mechanisms for concurrency or multi-tasking. Following power-on of the smartcard and initialization, the basic execution sequence for MULTOS is as follows:

- a) Wait for input from the IFD.
- b) Parse the input.
- c) If the input is a MULTOS command, process the command and write a response to the IFD.
- d) Otherwise, execute the currently selected application and write to the IFD any output created by the application.
- e) Loop back to a).

Applications to be loaded on MULTOS-based smartcards are written in a hardware-independent language called MULTOS Executable Language (MEL). MEL applications are interpreted by MULTOS, rather than being compiled and executed directly on the smartcard processor.

MULTOS also provides for shared code routines, called Codelets, which can be called by an executing application. Codelets can be loaded into MULTOS during IC manufacture or at smartcard personalization time. A codelet has its own code address space but executes in the context of the calling application, so has access to the application's data. MULTOS is targeted to operate on the Infineon Technologies IFX_CCI_000014 family Smartcard Integrated Circuits (ICs). The IC provides the microprocessor to execute the instructions comprising the executable code of MULTOS.

2.6.2 TOE functions

The two subsystems of the TOE (namely VM and MM) implement the following functions:

1. Managing the memory effects of the application life-cycle i.e. opening, loading, creation, selection, de-selection, exit and removal: the control of load and removal certificates/permissions is however not ensured by the TOE but by the CH (Command Handlers) subsystem.
2. Interpreting the MEL instructions and primitives contained in the loaded applications
3. Handling the interaction between loaded applications (by delegation mechanism)

The TOE functions enforce the access control policy of MULTOS. Indeed, the TOE maintains separate storage and execution space for applications loaded onto an MCD (MULTOS Carrier Device) (using function 1). The application execution mechanism (i.e. the functions 2 and 3) ensures that each application, including its code and data areas, is kept separate from other loaded applications. Each loaded application is restricted to its own code and data space and cannot gain un-authorized access to the code or data of another loaded application.

2.6.3 Intended Method of Use

MULTOS is intended to provide a hardware-independent environment for the execution of multiple applications that provide a variety of functions and services to the holder of the smartcard. Applications may be developed and supplied by different organizations from different industries, and consequently may provide many different services e.g., financial, communication or access control. The security requirements of different applications may also vary (i.e., some applications may require a high level of security while others may only have a low level or no security requirements).

A user of a MULTOS-equipped smartcard will be able to select any of the loaded applications and execute them. The user will access the facilities of the smartcard via an appropriate IFD. MULTOS implements a command interface for handling commands received from the IFD.

MULTOS provides a number of system calls (called primitives) which allow the currently executing application to request particular services from MULTOS.

MULTOS provides the following features:

- MULTOS will ensure all requests to load applications are appropriately authorized. MULTOS will support a capability to ensure the authenticity and integrity of an application when loading the application onto the smartcard. MULTOS will also ensure all requests to delete applications are appropriately authorized. Reasons for wishing to delete applications may be because they are found to contain errors, because an updated application is available, or to make room on the smartcard for a more desirable application.
- MULTOS will support a capability to load encrypted applications onto the smartcard, decrypt such applications and make them available to the smartcard user for execution
- MULTOS will ensure no application loaded on the smartcard can interfere with the operation of any other loaded application or with MULTOS. MULTOS will also ensure that an application's code and data will not be available to other applications after it has been deleted. MULTOS will provide the capability to authenticate a card as a valid MULTOS equipped smartcard.
- MULTOS will provide the capability to restrict the use of regulated features of the smartcard (e.g., strong cryptography) to authorized applications.
- MULTOS defines certain functions (installing keys, loading applications and deleting applications) as sensitive functions. For each of these functions, if the number of failed attempts to execute the function reaches a pre-defined limit over the life of the smartcard, MULTOS will permanently disable the function. In the case of installing keys, this means the card is unusable, as no applications can be loaded until keys have been installed. In the cases of application loading and deleting, other functions of the card remain available.

It is assumed that authorized applications which are loaded and executed by MULTOS are responsible for the secure processing of their own information. MULTOS provides an environment for secure loading and execution of smartcard applications.

The MULTOS access control policy maintains separate storage and execution space for applications loaded into an MCD (MULTOS Carrier Device). The application execution management mechanism ensures each application, including its code and data areas, is kept separate from every other application loaded on the MCD. Each application that is restricted to its own code and data space cannot gain access to the code or data of another loaded application.

2.6.4 Actors

Actors	Identification
Integrated Circuit (IC) Developer	IFX
Embedded Software Developer	Gemalto
Formal model and proof	Trusted Labs
Integrated Circuit (IC) Manufacturer	IFX
Initializer	Gemalto
Pre-personalizer	Gemalto
Personalization Agent	The agent who is acting on the behalf of the issuing State or Organization and personalize the MRTD (Machin Readable Travel Document) for the holder by activities establishing the identity of the holder with biographic data.
Card Holder	The rightful holder of the card for whom the issuer personalizes it.

Table 1: Identification of the actors

2.6.5 Smartcard Product Life Cycle

The Smartcard product life-cycle is decomposed into seven phases, according to the “Smartcard Integrated Circuit Protection Profile”.

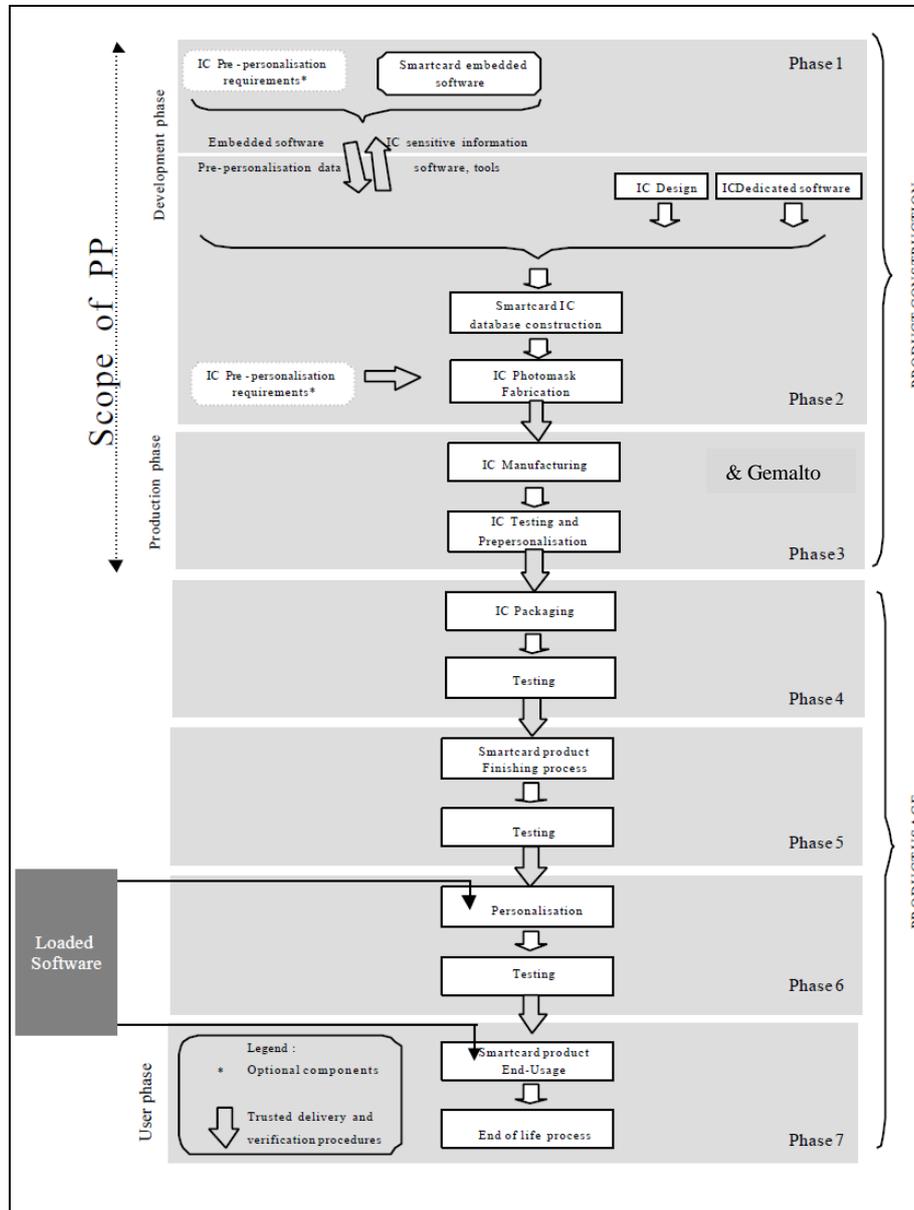


Figure 2: Smartcard IC with Multi-Application Platform Life Cycle

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

The IC does not contain any part of the IDMotion V2 software prior to phase 5. The loading of the IDMotion V2 software occurs during phase 5 in the flash memory, after which the IC Platform software loading service is locked and no more available (No patching process is possible after phase 5).

As described, at the end of phase 5 Gemalto delivers personalized IDMotion V2 product to our customers. At this stage, the TOE is entirely built and protects itself through the security mechanisms implemented in the operating system and the underlying IC. Consequently, the TOE delivery point - which determines the boundary between the ALC and AGD Common Criteria assurance classes - is put at the end of phase 5.

IDMotion V2 Virtual Machine Security Target

Notes related to Gemalto applications development

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

Note related to patch development

No patch is present within the TOE for the present evaluation. The patch mechanism is disabled at the end of the phase 5 (pre-personalization).

Evaluation scope: life-cycle boundary

Phase	Description / comments		Who	Where
1	IDMotion V2 platform development	Platform development & tests	Gemalto MULTOS R&D team secure environment	Gemalto Sydney Gemalto Fareham (UK)
			Gemalto SL Crypto team secure environment	Gemalto Meudon Gemalto La Ciotat
			Gemalto GBU R&D team secure environment	Gemalto Meudon Gemalto Singapore
	Formal development	Development of formal model and proof	Trusted Labs	Trusted Labs Meudon
	eTravel applet development	<ul style="list-style-type: none"> Applet development Applet tests 	Gemalto GBU R&D team secure environment	Gemalto Meudon
IAS applet development	<ul style="list-style-type: none"> Applet development Applet tests 	Gemalto GBU R&D team secure environment	Gemalto Singapore	
2	IC development	IC development	IC developer secure environment	IC development site(s) Refer to [CR-IC]
3	IC manufacturing	Manufacturing of virgin integrated circuits embedding a flash loader protected by a dedicated transport key.	IC manufacturer secure environment	IC manufacturing site(s) Refer to [CR-IC]
4	SC manufacturing: IC packaging, also called "assembly"	IC packaging & testing	Gemalto Production teams secure environment	Gemalto Gémenos Gemalto Singapore Gemalto Vantaa
5	SC manufacturing & pre-personalization	<ul style="list-style-type: none"> Module embedding in plastic card bodies Loading of the Gemalto software (platform and applets) SC initialization (profile building, loading of data needed for card pre-personalization...) 	Gemalto Production teams secure environment	Gemalto Gémenos Gemalto Singapore Gemalto Vantaa Gemalto Tczew Gemalto Curitiba Gemalto Montgomery
6	SC Personalization	Creation of files and loading of end-user data	SC Personalizer: Gemalto or another accredited company secure environment	SC Personalizer site
7	End-usage	End-usage for SC issuer	SC Issuer	Field
		End-usage for cardholder	Cardholder	Field

Table 2: Product and TOE life-cycle phases

IDMotion V2 Virtual Machine Security Target

Remark1: Initialization & pre-personalization operation could be done on module or on other form factor. The form factor does not affect the TOE security.

Remark2: Alternative life cycle, wafer are shipped by Infineon to form factor manufacturer (no module manufacturing required) and initialization /pre-personalization is done in Gemalto site.

Remark3: For initialization/pre-personalization IC flash loader could be used based the IC manufacturer recommendation.

Remark4: Embedding (module put on a dedicated form factor) will be done on an audited site defined in the phase 5 of the “Table 2: Product and TOE life-cycle phases”.

Remark5: The Phase 5 correspond to the IDMotion V2 Issuance process describe as below for the parts called “Flash Loading/Card Manufacturing” and “Card Pre-enablement”.

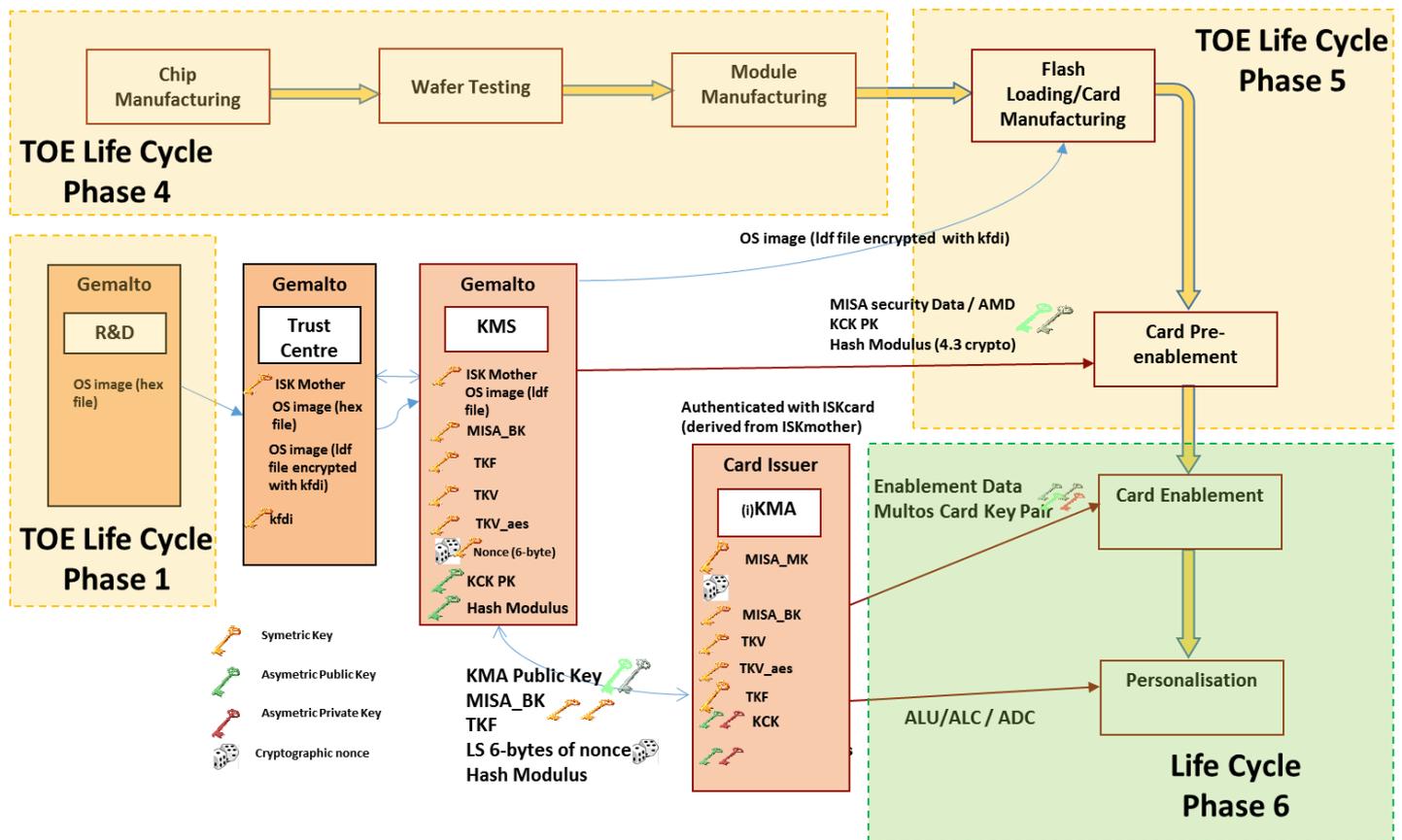


Figure 3: MULTOS Platform Life Cycle

2.6.6 Target of Evaluation Location and Usage

MULTOS will initially be developed in software. Following successful implementation and testing, the MULTOS executable will be masked in Flash Memory and embedded on smartcards, inlays/booklets, modules, or chips.

Once the MULTOS chip has been embedded, interaction with it will be via commands issued to smartcards (or inlays/booklets, modules, or chips) from an IFD or service requests (i.e., MULTOS system calls, known as primitives) made by an executing application.

2.6.7 Supporting Firmware

MULTOS requires firmware run-time libraries to support writing data to flash memory. These libraries are supplied by Infineon Technologies. They provide low-level routines to support writing data to flash memory, which is used on the target smartcard for the storage of applications. MULTOS requires the run-time libraries to execute correctly according to specification, to ensure data is written to the correct address within flash memory.

2.6.8 Supporting Security Infrastructure

It is assumed MULTOS-equipped smartcards and MULTOS applications will be manufactured and distributed within a commercial framework providing a procedural security infrastructure.

Figure 5 provides a simplified context diagram of the MULTOS commercial framework and security infrastructure.

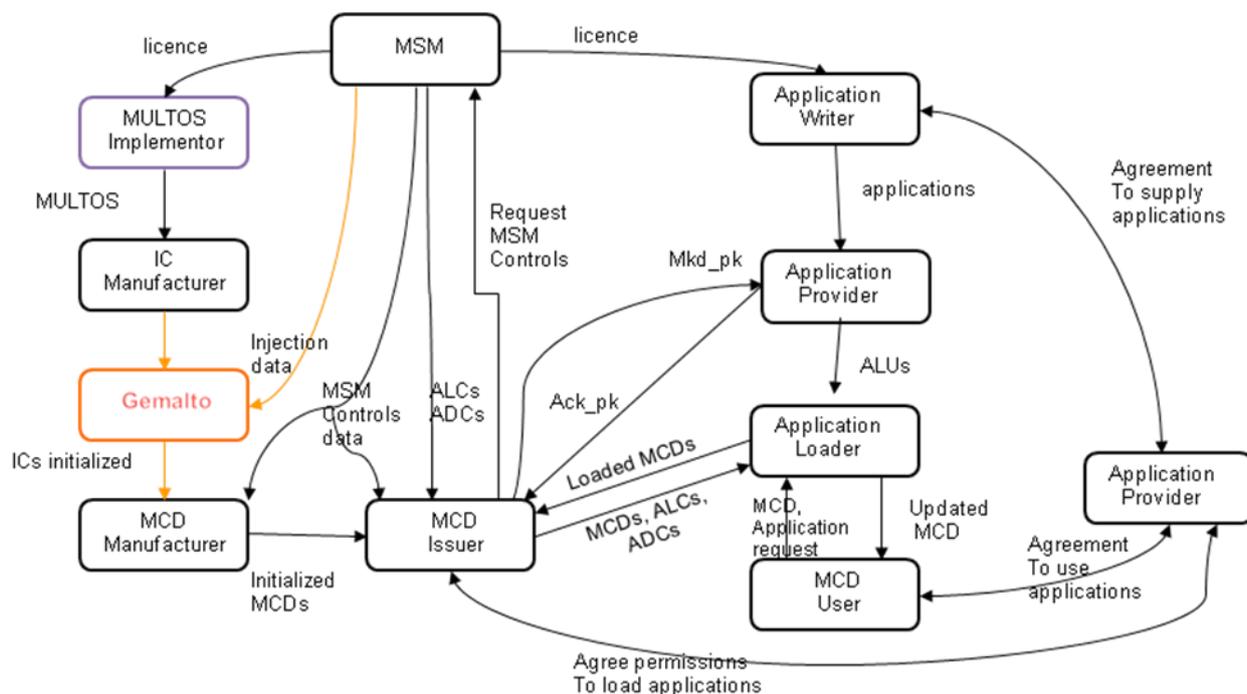


Figure 4: MULTOS Infrastructure Context Diagram

Legend

- MSM:** Multos Security Manager
- MCD:** Multos Carrier Device
- ALU:** Application Load Unit
- ALC:** Application Load Certificate
- ADC:** Application Delete Certificate
- Mkd_pk:** Held by MSM; copy provided to Application Providers; used by Application Providers to encrypt KTU for target MCD.
- Ack_pk:** Provided to MCD Issuer, who gets it certified by the MSM when ALCs and ADCs are requested

IDMotion V2 Virtual Machine Security Target

In this product version including Gemalto the role of the IC Manufacturer is split.

- The founder (Infineon) manufactures the chips and performs limited FLASH Memory injection including a diversified transport key (used by Initialization Mode).
- Gemalto then performs the final manufacturing step by injecting the iKMA keys and data into the FLASH memory.

The following roles and responsibilities are assumed within the infrastructure:

- MULTOS Security Manager (MSM):** defines and polices the MULTOS security infrastructure and provides criteria and services necessary for MULTOS participants to operate within the infrastructure. It acts as Certification Authority for the security infrastructure. It is assumed only one MSM exists. The MSM must be trusted by all participants in the infrastructure
- MULTOS Implementor:** the organization that implements a MULTOS version.
The MULTOS Implementor is licensed by MAOSCO and provides its MULTOS version to the IC Manufacturer. The MULTOS Implementor requests the MSM to provide MSM Controls Data, although this may be delivered to the MCD Manufacturer or MCD Issuer.
- Integrated Circuit (IC) Manufacturer:** manufacturer of silicon from which chips and smartcards are made. It is assumed the IC Manufacturer is trusted to perform its tasks correctly: This includes:
 - To perform limited Flash memory injection,
 - The injection of diversified transport key (used by Initialization Mode).
- Gemalto:** is included in the new scheme in order to perform the final manufacturing step by injecting the iKMA keys and data into the Flash memory;

The initialized ICs are provided to MCD Manufacturers:

- MULTOS Carrier Device (MCD) Manufacturer:** responsible for embedding the IC in its plastic carrier and for background printing on the card. The result is an initialized MCD. This operation is assumed not to be security sensitive. The MCD Manufacturer may also receive MSM Controls Data from the MSM and enable the MCDs. Initialized and enabled MCDs are provided to MCD Issuers.
- MCD Issuer:** responsible for issuing to users the MCD itself. The MCD Issuer may also enable initialized MCDs, by loading MSM Controls Data received from the MSM onto the MCDs. MCD Issuers retain the ultimate authority over what applications are loaded on their MCDs. MCD Issuers register applications with the MSM, provide information related to the applications and receive application load and delete certificates from the MSM.
- Application Writer:** licensed by MAOSCO to produce applications for MULTOS. Supplies applications under contract to Application Issuers.
- Application Issuer:** an organization that wishes to offer an application to MCD Users.
The Application Issuer agrees with an MCD Issuer that the application can be loaded onto MCDs belonging to the MCD Issuer.
- Application Provider:** the organization that takes responsibility for an application, by certifying it with the organization's public key and encrypting it where necessary. The Application Provider is a role that can be performed by an Application Writer, Application Issuer or MCD Issuer, rather than necessarily, being an organization in its own right.
- Application Loader:** responsible for performing the technical operation of loading applications onto MCDs. The Application Loader enters into an agreement with one or more Application Issuers and MCD Issuers for loading applications supplied by one or more Application Providers.
- MCD User:** final user of the MCD.

The MSM authorizes potential MULTOS platforms. To receive MSM authorization, a platform must comply with criteria covering attributes of the platform itself and the procedures associated with its manufacture.

MULTOS platforms are assumed to satisfy the following requirements:

- a) They are manufactured in a controlled environment conforming to MSM rules.
- b) They are subject to type approval by the MSM.
- c) They possess a level of tamper resistance.

2.6.9 Application Load Units (ALU)

An Application Load Unit (ALU) is generated by an Application Provider to load applications. An ALU may be uncertified or certified. An uncertified ALU simply contains a clear text copy of the application. A certified ALU contains, in addition to the application, an application signature, which authenticates the application. The Application Provider may also encrypt parts of the application, in which case a Key Transformation Unit is included in the certified ALU.

2.6.10 Key Transformation Unit (KTU)

An Application Provider wishing to utilize application confidentiality will generate a Key Transformation Unit (KTU). The KTU contains descriptors for the areas of the application's code and data that have been encrypted. Each descriptor contains the start address of the protected area, the length of the protected area, an indicator of the algorithm used and the key used to encrypt the contents of the area. The descriptors and some header information (including application identifier and target MCD number) are then encrypted, using the target MCD's public transport key, and included in the KTU.

2.6.11 Application Load and Delete Certificates (ALCs & ADCs)

Application Load Certificates (ALCs) and Application Delete Certificates (ADCs) are generated by the MSM to respectively load and delete an application on to and from an MCD. Each ALC contains the unique Application ID of the application for which it is created. Each ALC refers to a particular domain, which defines the set of MCDs that the application may be loaded on to and deleted from.

The domain is defined by a set of load permissions and may be:

- a) A specific MCD.
- b) A subset of the cards issued by an MCD Issuer.
- c) All cards issued by an MCD Issuer.
- d) Limited to a subset of cards enabled on specific dates.
- e) A combination of the above.

An ALC contains load controls that define exactly what load operations are allowed.

The load controls specify:

- a) If application signature has been used.
- b) If application confidentiality has been used.
- c) If reloading a deleted application is permitted.

The ALC also contains feature permissions, which define what regulated features the application may use:

- Access flags for access to MULTOS functions:

- Cryptographic access
- Allow access to particular interfaces (contact/contactless)
- Allow access to shared PIN
- Allow access to card block/unblock primitives
- ...

- Application Permissions - used to match ALC with ALU & Card, or ADC to Card – plus one-time load (history)

The ADC for an application is created at the same time as the ALC. It contains the same unique Application ID and the same set of load permissions as the corresponding ALC.

IDMotion V2 Virtual Machine Security Target

2.6.12 Keys

The following table lists each of the cryptographic keys required to support the MULTOS security infrastructure. Each key is identified by a name, the key type (symmetric/asymmetric) and its role within the MULTOS security infrastructure. Asymmetric keys have two components: a secret and a public key. In the following table, secret components of asymmetric keys are identified by a “_sk” suffix, and public components by the suffix “_pk”.

Key name	Key definition	Key part	Role
MISA_mk <i>symmetric</i>	MISA Master Key		Generated by MSM; used by MSM to generate MISA_bk
MISA_bk <i>symmetric</i>	MISA Base Key		Each key value is unique to a given MISA. Used by MISA and MSM to determine TKV for a specific MCD These keys are diversified during wafer production to inject each chip with a unique transport key (TKV)
TKV <i>symmetric</i>	MCD-specific transport key		Generated by MSM; stored in FLASH memory of target MCD; used by MSM to encrypt MCD-specific MSM Controls Data and also by MULTOS to decrypt the MSM Controls Data. Unique key used to encrypt enablement data
TKF <i>symmetric</i>	Fixed part of MCD-specific transport key		Generated by MSM; stored in FLASH memory of MCD; used by MSM, MCD Issuer and Application Loader to check authenticity of target MCD; TKF is fixed for all MCDs
MKD <i>asymmetric</i>	MCD-specific asymmetric transport key.	mkd_sk	Held in FLASH memory of target MCD; used by MULTOS to decrypt KTU (Key Translation Unit)
		mkd_pk	Held by MSM; stored in FLASH memory of target MCD; copy provided to Application Providers; used by Application Providers to encrypt KTU (Key Translation Unit) for target MCD
		mkd_pk_c	mkd_pk, certified by MSM using tkck_sk to indicate its authenticity. By decrypting this with tkck_pk the mkd_pk can be recovered for use
TKCK <i>asymmetric</i>	Transport Key Certification Key	tkck_sk	Held securely by MSM; used by MSM to certify MCD-specific public transport keys (mkd_pk) Used to sign the mkd_pk during enablement data generation
		tkck_pk	Held by MSM; copy provided to Application Providers; used by Application Providers to verify and retrieve certified MCD-specific public transport keys (mkd_pk_c)
DEK <i>symmetric</i>	Data Encryption Key		Used to encipher sections of the application. It is included as part of the KTU, which is secured by the chip's key (MKD).
ACK <i>asymmetric</i>	Application Provider's asymmetric key <i>Generated by Application Provider</i>	ack_sk	Held by Application Provider; used by Application Provider to sign application certificate
		ack_pk	Provided to MCD Issuer, who gets it certified by the MSM when ALCs and ADCs are requested
HM <i>asymmetric</i>	Hash Modulus		While not strictly a key as such, this RSA public key is used as an input the MULTOS proprietary Asymmetric Hash algorithm which is based on RSA. This is used during the verification of ALC/ADCs, MSM controls and application signatures The secret part of the key is not use
KCK <i>asymmetric</i>	Global Key Certification Key <i>Generate by MSM</i>	kck_sk	Held securely by MSM; used by MSM to certify ADCs and ALCs (and indirectly through these, Application Provider public keys (ack_pk)); Guarantees the authenticity of applications
		kck_pk	Held in FLASH memory of every MCD ; used by MULTOS to verify ALCs, ADCs and Application Provider public keys

Table 3: MULTOS Security Infrastructure Keys

The critical keys, which are managed by the MSM and support the MULTOS security infrastructure, are:

- a) Global Key Certification Key (KCK)
- b) Transport Key Certification Key (TKCK)

The KCK supports the authentication of MULTOS applications and the authorization of requests by MCD Issuers to load and delete applications. The secret KCK (kck_sk) is held securely by the MSM and is used to sign ALCs and ADCs. ALCs and ADCs contain the Application Provider's public key (ack_pk), so signing the ALC/ADC also certifies ack_pk for use with MCDs. The public KCK (kck_pk) is installed in the Flash memory of each instance of MULTOS (i.e., it is available on every MCD).

The TKCK supports the provision of application confidentiality and MCD authentication. An asymmetric transport key is created for each MCD (this is MKD). The public part of MKD (mkd_pk) is certified by the MSM using the secret part of the TKCK (tkck_sk). Application Providers wishing to utilize application confidentiality when loading applications onto an MCD obtain from the MCD Issuer the public part of MKD, certified by the MSM (i.e., mkd_pk_c). The Application Provider uses the public part of the TKCK (tkck_pk) to authenticate mkd_pk and uses mkd_pk to encrypt the KTU for the target MCD.

2.6.13 MULTOS Initialization Security Data (Card Pre-enablement)

The MISA (MULTOS Injection Security Application) Security Data contains the unique identity and transport keys used to identify and protect the device during distribution. The security of a MULTOS device and the MULTOS scheme depends upon this data.

MULTOS Initialization Security Data is generated by the MSM and supplied to Gemalto for incorporation into MULTOS. MULTOS Initialization Security Data comprises the security data, which are injected.

The MULTOS security data is injected during MULTOS initialization.

This is performed using a device called a MULTOS Injection Security Application (MISA).

The MSM constructs data for each MISA, including a unique MISA identifier.

Two crypto schemes can be used:

- 2 keys multi-key DES to protect a MULTOS device.
- AES to protect devices during distribution.

It will be necessary for a manufactured device to either use the DES or AES crypto scheme and for this to be identified and decided at manufacturing.

IDMotion V2 Virtual Machine Security Target

The MISA then constructs the data to be injected into the MCD.

Name	Definition	Description	
security_data	jump_code	Internal use	
	random_number		
[security_data_des_key]	msm_controls_algorithm_id	Determines whether the security_data_des_key or security_data_aes_key structure is used and present in the MISA security data structure.	
	[security_data_des_key] or [security_data_aes_key]		
	tkf	MCD-specific symmetric transport keys (TKF and TKV), which are used in loading the MCD-specific asymmetric transport key (MKD) as a component of the MSM Controls Data	
	tkv		
	mism_id	Two bytes ID that identify the keyset used by the card Keyset used to generate the TKV	
	icc_serial_number	A unique identifier based on the MISA identifier and ICC serial number	
	initialisation_date	Initialization date, indicating when the security data was injected into the MCD	
	security_level	A security flag indicating MSM Controls Data has not been loaded. (Redundancy usage with OS)	
	random_number		
	[security_data_aes_key]	tkf	MCD-specific symmetric transport keys (TKF and TKV), which are used in loading the MCD-specific asymmetric transport key (MKD) as a component of the MSM Controls Data
		tkv_aes	
		tkv_aes_length	
		mism_id	Two bytes ID that identify the keyset used by the card Keyset used to generate the TKV
		icc_serial_number	A unique identifier based on the MISA identifier and ICC serial number
initialisation_date		Initialization date, indicating when the security data was injected into the MCD	
security_level		A security flag indicating MSM Controls Data has not been loaded. (Redundancy usage with OS)	
random_number			

Table 4: MISA security Data

Figure 5 depicts the information flow from the MSM to the Gemalto.

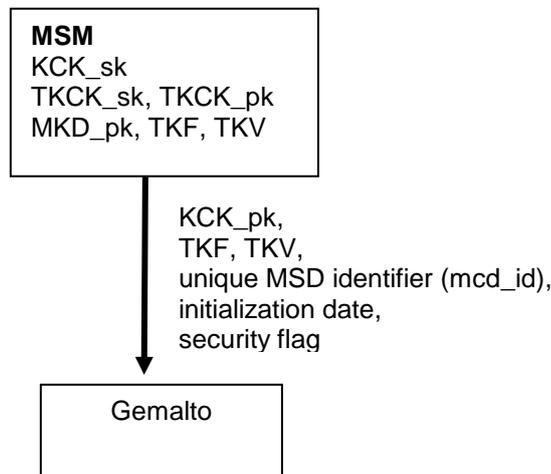


Figure 5: MULTOS Initialization Security Data Information Flow

2.6.14 MSM Controls Data (Card enablement)

The MCD must be loaded with its permissions and asymmetric transport key set (i.e., MKD) before application loading can be supported. The transport key (comprising the private key, and certified public key) and permissions are provided by the MSM to the MCD Manufacturer or MCD Issuer in MSM Controls Data. This data also includes the MCD's unique identifier and is protected by the MCD-specific symmetric transport key (TKV). Once the transport keys have been generated and encrypted, MSM destroys the copy of mkd_sk it generated, in order to ensure the confidentiality of this key.

Figure 6 depicts the information flow from the MSM to the MCD Manufacturer or MCD Issuer.

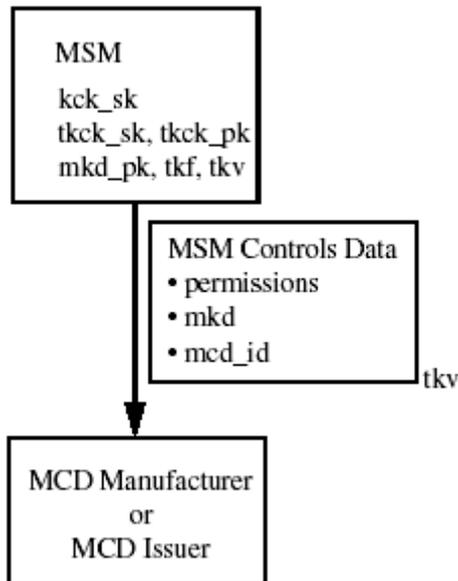


Figure 6: MSM Controls Data Information Flow

2.6.15 Loading Applications (Personalisation)

The principal key and data exchanges involved in loading applications onto MCDs are depicted in Figure 7.

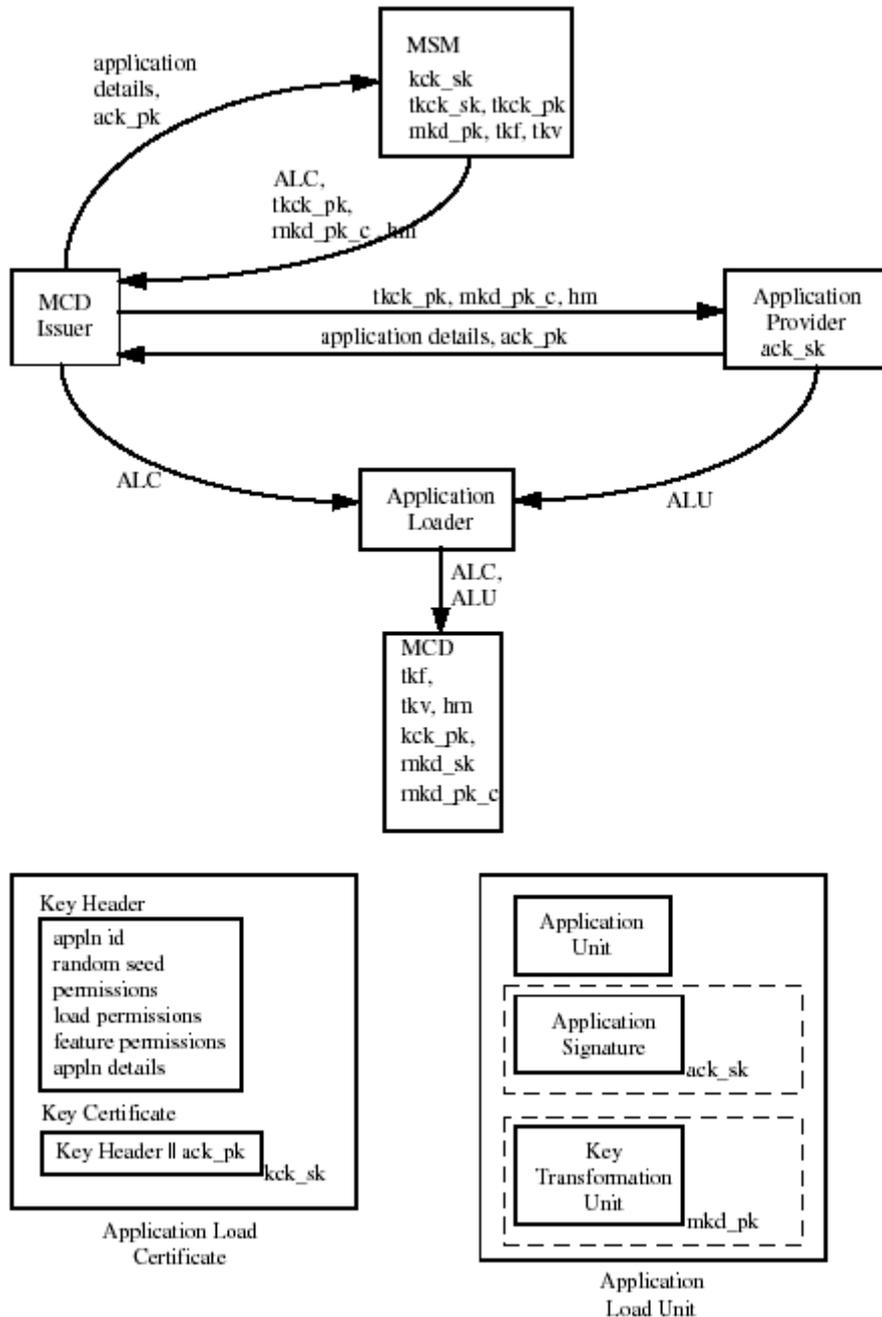


Figure 7 : Principal Key and Data Exchanges in Loading MCD Applications

The Application Provider provides its public key (`ack_pk`) and details of the application to be loaded to the MCD Issuer. The MCD Issuer forwards `ack_pk` and the application details to the MSM. The MSM creates an ALC, which contains the application details in the Key Header. MSM creates the Key Certificate over the information in the Key Header, concatenated with `ack_pk`, and signs it with the secret KCK (`kck_sk`).

IDMotion V2 Virtual Machine Security Target

The MSM provides the ALC to the MCD Issuer. If the Application Provider has requested use of application confidentiality, the MSM also provides the MCD Issuer with the target MCD's certified public transport key (mkd_pk_c) and the public TKCK (tkck_pk).

The MCD Issuer provides mkd_pk_c and tkck_pk to the Application Provider and the ALC to the Application Loader.

The Application Provider creates an ALU for the application to be loaded onto the MCD.

The ALU comprises the following components:

- a) Application unit, containing the application's code and data.
- b) Application signature (optional).
- c) KTU (optional).

If the Application Provider requires application authentication, it includes an application signature in the ALU. The application signature is created over the application unit and signed with the Application Provider's secret key (ack_sk).

If the Application Provider requires application confidentiality, it includes a KTU. The KTU is signed using the target MCD's public key (mkd_pk), retrieved from mkd_pk_c using tkck_pk.

The Application Provider provides the ALU to the Application Loader. The Application Loader loads the ALU on the target MCD, using the ALC to demonstrate the load has been authorized by the MSM.

3. CONFORMANCE CLAIMS

This section describes how the ST claims conformance with Common Criteria for Information Technology Security Evaluation Version 3.1, Revision 5.

3.1 COMMON CRITERIA CONFORMANCE CLAIMS

This ST has been built with Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, as the following:

- Part 2 conformant.
- Part 3 conformant with EAL7 level.

The EAL5 level from CC Part 3 is augmented with the assurance components ALC_DVS.2 and AVA_VAN.5.

3.2 PROTECTION PROFILE CLAIM AND PACKAGE CLAIM

This ST is based on PP/0010 Version 2.0, Issue November 2000, registered at the French Certification Body. Please note that the PP/0010 is upwardly compatible with the PP/9806 and PP/9911. Therefore, this ST is also based on Smartcard IC Protection Profile PP/9806, Version 2.0, Issue September 1998 and Smartcard IC with Embedded Software Protection Profile PP/9911, Version 2.0, issued in June 1999.

Note: Items which are common to PP/9806 and PP/0010 are indicated by a “*” in this Security Target.

4. SECURITY PROBLEM DEFINITION

This section describes the security problem to be addressed by the TOE and the operational environment in which the TOE is intended to be used. It provides a description of the assets to be protected, the threats, the organizational security policies and the assumptions about the operational environment of the TOE.

4.1 ASSETS

Assets are security relevant elements of the TOE that include:

Assets linked to the IC with Multi-Application Secure Platform itself:

- The IC specifications, design, development tools.
- The IC Dedicated software.
- The integrity of the Multi-Application Platform Software.
- Multi-Application Platform specifications, implementation, test programs and related documentation.
- The confidential TSF data (tkf, tkv and mkd_sk).

Assets linked to Loaded-Applications on the platform:

- Application provider User Data:
 - Loaded-Application software loaded on the platform.
 - Confidential Loaded-Application SF data. (Encrypted SF data for the eventual Loaded Application Security Functions).
- The TOE resources:
 - Card resources: memory space and computation power made available to a Loaded-Application and its security functions.

Assets linked to end user, card holder and application provider:

- End User Data for users of Native Applications.
- End User Data for users of Loaded Applications.

NOTE: even if the PP scope does not include the applications, the TOE must provide security mechanisms such that Loaded Applications can protect the End User data when required.

Assets have to be protected in terms of confidentiality, authenticity and control of their origin.

4.2 THREATS

The TOE and its operational environment as defined in chapter 2 are required to counter the threats described hereafter.

A threat agent (an attacker) wishes to abuse the assets either by functional attacks or by environmental manipulation, by specific hardware manipulation, by a combination of hardware and software manipulations or by any other type of attacks.

Threats have to be split in:

- Threats against which specific protection within the TOE is required (class I).
- Threats against which specific protection within the environment is required (class II).

4.2.1 Unauthorized Full or Partial Cloning of the Target of Evaluation

T.CLON*

Functional cloning by attackers of the TOE (full or partial) appears to be relevant to all phases of the TOE life-cycle, from phase 1 to phase 7, but only phases 1 and 4 to 7 are considered here, since functional cloning in phases 2 and 3 are purely in the scope of Smartcard IC PP. Generally, this threat is derived from specific threats by attackers, addressing User Data and potentially TSF data, combining unauthorized disclosure, modification or theft of assets at different phases.

4.2.2 Threats on Phase 1

Common Criteria v3 does not require threats for the development environment so these threats (and any references to them) should be ignored for this version of Common Criteria. Instead, Common Criteria v3 requires that the development environment is evaluated in the ALC assurance class of the evaluation.

During phase 1, three types of threats by attackers have to be considered:

- a) Threats on the Smartcard Embedded Software (ES) and its development environment, such as unauthorized disclosure, modification or theft of the Smartcard Embedded Software and/or initialization data.
- b) Threats on the assets transmitted from the IC designer to the Smartcard software developer during the Smartcard ES development.
- c) Threats on the Smartcard Embedded Software and initialization data transmitted during the delivery process from the Smartcard software developer to the IC designer.

Unauthorized disclosure of assets

This type of threat covers unauthorized disclosure of assets by attackers who may possess a wide range of technical skills, resources and motivation. Such attackers may also have technical awareness of the product.

T.DIS_INFO* (type b)

An attacker may cause unauthorized disclosure of the assets delivered by the IC designer to the Smartcard Embedded Software developer, such as sensitive information on IC specification, design and technology, software and tools if applicable.

T.DIS_DEL* (type c)

An attacker may cause unauthorized disclosure of the Asset Smartcard Embedded Software and any additional *application data* (such as IC pre-personalization requirements) during the delivery to the IC designer.

NOTE application data means TSF data.

T.DIS_ES1 (type a)

An attacker may cause unauthorized disclosure of ES (technical or detailed specifications, implementation code) and/or TSF data (such as secrets, or control parameters for protection system, specification and implementation for security mechanisms).

T.DIS_TEST_ES (type a and c)

An attacker may cause unauthorized disclosure of the Smartcard ES test programs or any related information.

Theft or unauthorized use of assets

Potential attackers may gain access to the TOE and perform operations for which they are not authorized. For example, such an attacker may personalize, modify or influence the product in order to gain access to the Smartcard application system.

T.T_DEL* (type c)

An attacker may target theft of the Smartcard Embedded Software and any additional *application data* (such as pre-personalization requirements) during the delivery process to the IC designer.

NOTE application data means TSF data.

T.T_TOOLS (type a and b)

IDMotion V2 Virtual Machine Security Target

An attacker may target theft or unauthorized use of the Smartcard ES development tools (such as PC, development software, databases).

T.T_SAMPLE2 (type a)

An attacker may target theft or unauthorized use of TOE samples (e.g. bond-out chips with the Embedded Software).

Unauthorized modification of assets

The TOE may be subjected by attackers to different types of logical or physical attacks, which may compromise security. Due to the intended usage of the TOE (the TOE environment may be hostile), the TOE security may be bypassed or compromised reducing the integrity of the TOE security mechanisms and disabling their ability to manage the TOE security. This type of threats includes the implementation of malicious Trojan horses.

T.MOD_DEL* (type c)

An attacker may cause unauthorized modification of the Smartcard Embedded Software and any additional *application data* (such as IC pre-personalization requirements) during the delivery process to the IC designer.

Note: Application data means TSF data.

T.MOD (type a)

An attacker may cause unauthorized modification of ES and/or TSF data or any related information (technical specifications).

4.2.3 Threats on Delivery for/from Phase 1 to Phases 4 to 6

Threats by attackers on data transmitted during the delivery process from the Smartcard developer to the IC packaging manufacturer, the Finishing process manufacturer or the Personaliser.

These threats are described hereafter:

T.DIS_DEL1

An attacker may cause unauthorized disclosure of and ES personalization Data during delivery to the IC Packaging manufacturer, the Finishing process manufacturer or the Personaliser.

T.DIS_DEL2

An attacker may cause unauthorized disclosure of ES personalization Data delivered to the IC Packaging manufacturer, the Finishing process manufacturer or the Personaliser

T.MOD_DEL1

An attacker may cause unauthorized modification of ES personalization Data during delivery to the IC Packaging manufacturer, the Finishing process manufacturer or the Personaliser.

T.MOD_DEL2

An attacker may cause unauthorized modification of and ES personalization Data delivered to the IC Packaging manufacturer, the Finishing process manufacturer or the Personaliser.

4.2.4 Threats on Phases 4 to 7

During these phases, the assumed threats could be described in four types:

- Unauthorized disclosure of assets.
- Theft or unauthorized use of assets.
- Unauthorized modification of assets.
- Threats on Loaded-Applications.

Unauthorized disclosure of assets

This type of threat covers unauthorized disclosure of assets by attackers who may possess a wide range of technical skills, resources and motivation. Such attackers may also have technical awareness of the product.

T.DIS_ES2

IDMotion V2 Virtual Machine Security Target

An attacker may cause unauthorized disclosure of ES, Native-Application, and Loaded-Application TSF Data (such as data protection system, memory partitioning, cryptographic programs and keys).

Theft or unauthorized use of assets

Potential attackers may gain access to the TOE and perform operation for which they are not allowed. For example, such attackers may personalize the product in an unauthorized manner, or try to gain fraudulently access to the Smartcard system.

T.T_ES

An attacker may cause unauthorized use of TOE. (e.g. bond out chips with embedded software).

T.T_CMD

An attacker may cause unauthorized use of instructions or commands or sequence of commands sent to the TOE.

Unauthorized modification of assets

The TOE may be subjected by attackers to different types of logical or physical attacks, which may compromise security. Due to the intended usage of the TOE (the TOE environment may be hostile), the TOE security parts may be bypassed or compromised reducing the integrity of the TOE security mechanisms and disabling their ability to manage the TOE security. This type of threat includes the implementation of malicious Trojan horses, Trapdoors, downloading of viruses or unauthorized programs.

T.MOD_TSF

An attacker may cause unauthorized modification or destruction of TOE Security Function Data by any mean including probing, electronic perturbation etc.

T.MOD_LOAD

An attacker may cause unauthorized loading of Native Applications. This includes also illegal modification of eventual Native Applications. As the TOE described in a Security Target claiming this PP must include eventual Native Applications, their loading or modification must be blocked during the usage phase. The threat includes bypassing this blocking.

T.MOD_EXE

An attacker may cause unauthorized execution of Platform or application software.

T.MOD_SHARE

An attacker may cause unauthorized modification of Platform or application behavior by interaction of different programs.

T.MOD_SOFT*

An attacker may cause unauthorized modification of Smartcard Embedded Software and data.

4.2.5 Threats on Phases 6 to 7

Threats on assets linked to Loaded-Applications

These threats by attackers are specific to the Multi-Application Platform. They are centered on threats by attackers to loading/unloading of Loaded-Applications and to threats using a Loaded-Application to attack another.

T.LOAD_MAN

Attackers load an application on the platform by bypassing the Administrator. This threat could lead to undue usage of card resources, and for unverified application to attack on other Loaded-Application TSF or User data.

T.LOAD_APP

Attackers load an application that purports to be another Loaded-Application. This attacks card resources and end user data.

T.LOAD_OTHER

IDMotion V2 Virtual Machine Security Target

Attackers loading the software representation of a Loaded-Application intended for a specific platform domain onto other platform domains, thus taking from the Loaded-Application representation the security feature of being confined to a specific domain. This is an attack on Loaded-Application User Data.

T.LOAD_MOD

Attackers intercepting application load units and altering code or data without the permission of the Loaded-Application Provider. This attacks application provider user data.

T.APP_DISC

Attackers intercepting application load units and gaining access to confidential code or data. This is an attack on application provider user data's confidentiality and knowledge.

Note: T.APP_DISC is also present during phase A1.

T.APP_CORR

Attackers load an application that partially or completely overwrites another Loaded-Applications, either corrupts or gains access to code or data. This is an attack on Application Provider user data.

T.APP_REMOVE

Attackers remove a Loaded -Application without the involvement of the Administrator. This is an attack on Application Provider user data.

T.ERR_REMOVE

Attackers removing a Loaded-Application leaving confidential data and/or code in memory which can be examined This is an attack on Application Provider user data.

T.DEL_REMOVE

Attackers remove a Loaded-Application at the same time deleting part or all of another Loaded-Application. This is an attack on Application Provider user data.

T.APP_READ

Attackers use a loaded application to read confidential data or code belonging to another Loaded-Application. This attacks the confidentiality of User Data.

T.APP_MOD

Attackers use a Loaded-Application to modify data or code belonging to another Loaded-Application without its authorization. This is an attack on Application Provider user data (and also End User data).

T.RESOURCES

Attackers target total or partial destruction of card resources delivered by the platform.

4.2.6 Threats on Phase 7

Unauthorized disclosure of assets

T.DIS_DATA

Attackers may cause unauthorized disclosure of User (application provider and end user) data and TSF data.

Unauthorized modification of assets

T.MOD_DATA

Attackers may cause unauthorized modification or destruction of User (application provider and end user) Data and TSF data.

Table 2 given below indicates the relationship between the phases of the Smartcard life cycle, the threats and the type of the threats:

Threats	Phase 1	Phase A1	Phase 4	Phase 5	Phase 6	Phase 7
T.CLON*	Class II		Class I	Class I	Class I	Class I
T.DIS_INFO*	Class II					

IDMotion V2 Virtual Machine Security Target

Threats	Phase 1	Phase A1	Phase 4	Phase 5	Phase 6	Phase 7
T.DIS_DEL*	Class II					
T.DIS_DEL1	Class II		Class II	Class II	Class II	
T.DIS_DEL2			Class II	Class II	Class II	
T.DIS_ES1	Class II					
T.DIS_TEST_ES	Class II					
T.DIS_ES2			Class I	Class I	Class I	Class I
T.T_DEL*	Class II					
T.T_TOOLS	Class II					
T.T_SAMPLE2	Class II					
T.T_ES			Class I	Class I	Class I	Class I
T.T_CMD			Class I	Class I	Class I	Class I
T.MOD_DEL*	Class II					
T.MOD_DEL1	Class II		Class II	Class II	Class II	
T.MOD_DEL2			Class II	Class II	Class II	
T.MOD	Class II					
T.MOD_TSF			Class I	Class I	Class I	Class I
T.MOD_SOFT*			Class I	Class I	Class I	Class I
T.MOD_LOAD			Class I	Class I	Class I	Class I
T.MOD_EXE			Class I	Class I	Class I	Class I
T.MOD_SHARE			Class I	Class I	Class I	Class I
T.DIS_DATA						Class I
T.MOD_DATA						Class I
T.LOAD_MAN					Class I	Class I
T.LOAD_APP					Class I	Class I
T.LOAD_OTHER					Class I	Class I
T.LOAD_MOD					Class I/II	Class I/II
T.APP_DISC		Class II			Class I/II	Class I/II
T.APP_CORR					Class I	Class I
T.APP_REMOVE					Class I	Class I
T.ERR_REMOVE					Class I	Class I
T.DEL_REMOVE					Class I	Class I
T.APP_READ					Class I	Class I
T.APP_MOD					Class I	Class I
T.RESOURCES					Class I	Class I

Table 5: Relationship between phases and threats

Note: Phases 2 and 3 are covered in the scope of Smartcard IC PP.

4.3 ORGANIZATIONAL SECURITY POLICIES

OSP.CIPHER The TOE must contribute and provide cryptographic functions are required to actually protect the exchanged information. These cryptographic algorithms need to be consistent with cryptographic usage policies and standards. Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them.

OSP.CONF-ALU The confidential ALU includes a KTU (key transformation unit) used to encrypt sensitive sections of the ALU (or the entire ALU). KTU itself is encrypted off-card using the card's public asymmetric transport key (MKD-PK). MULTOS decrypts it using its private asymmetric transport key (MKD-SK).

4.4 ASSUMPTIONS

Security always concerns the whole operational environment of the TOE. The weakest element of the chain determines the total system security. Assumptions described hereafter must be considered for a secure system using Smartcard products.

4.4.1 Assumptions on the Target of Evaluation Delivery Process (Phases 4 to 7)

Procedures shall guarantee the control of the TOE delivery and storage process and conformance to its objectives as described in the following assumptions:

A.DLV_PROTECT*

Procedures shall ensure protection of TOE material/information under delivery and storage.

A.DLV_AUDIT*

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage.

A.DLV_RESP*

Procedures shall ensure that people dealing with the procedure for delivery have got the required skill.

4.4.2 Assumptions on Phases 4 to 6

A.USE_TEST*

It is assumed that appropriate functionality testing of the TOE is used in phases 4, 5 and 6.

A.USE_PROD*

It is assumed that security procedures are used during all manufacturing and test operations through phases 4, 5, 6 to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

4.4.3 Assumption on Phase 7

A.USE_DIAG*

It is assumed that secure communication protocols and procedures are used between Smartcard and terminal.

4.4.4 Assumption on Loaded-Application Development (Phase A1)

A.APPLI_CONT

Whenever a Loaded-Application is to be loaded on the platform, it is assumed that its development and production follow the Administrator Guidance.

4.4.5 Additional assumptions

The following assumptions are added because the corresponding TOE security objectives in [ST_full_IDMotionV2] is transferred to the TOE environment in this ST.

A.TAMPER_ES

Tampering with the security critical parts is prevented by the security mechanisms (in particular the unauthorized change of functional parameters, security attributes and secrets such as the lifecycle sequence flags and cryptographic keys).

A.SIDE

The interpretation of electrical signals from the hardware part is avoided.

A.CLON

The product functionality must be protected from cloning.

A.OPERATE

The continued correct operation of the security functions is assumed.

A.DIS_MECHANISM2

The embedded system security mechanisms are protected against unauthorized disclosure.

A.DIS_MEMORY

The sensitive information stored in memories is protected against unauthorized disclosure.

A.MOD_MEMORY

The sensitive information stored in memories is protected against corruption or unauthorized modification.

A.LOAD

Only application with permission of the administrator is loaded onto the platform.

A.SECURITY

The application load process must be able to guarantee, when required, the integrity, confidentiality, and to verify the claimed origin of the Loaded-Application code and data.

A.REMOVE

Removal of a Loaded-Application and consequent reuse of the Loaded-Application space is only to be performed with the authorization of the administrator. The space must not hold any information relative to data or code linked to the removed Loaded-Application.

A.CIPHER

There is a means to cipher sensitive data for application in a secure way. Supported cryptographic algorithms must be consistent with cryptographic usage policies and standards.

A.DECIPHER

There is a means to decipher the KTUs.

4.5 COMPOSITION TASKS

4.5.1 Statement of Compatibility – Threats

The following table lists the relevant threats of the IFX_CCI_000014 and provides the link to the threats related to the composite product, showing that there is no contradiction between the two.

IC Relevant Threat Label	IC Relevant Threat Title	IC Relevant Threat Content	Link to the composite-product threats
T.Phys-Manipulation	Physical Manipulation	An attacker may physically modify the Security IC in order to(i)modify User Data (ii) modify the Security IC Embedded Software (iii) modify or deactivate security services of the TOE, or (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.	T.DIS_ES2 T.T_CMD T.MOD_TSF T.MOD_EXE T.LOAD_MAN T.LOAD_APP T.APP_CORR T.APP_REMOVE T.ERR_REMOVE T.DEL_REMOVE

IDMotion V2 Virtual Machine Security Target

IC Relevant Threat Label	IC Relevant Threat Title	IC Relevant Threat Content	Link to the composite-product threats
			T.APP_READ T.APP_MOD T.DIS_DATA T.MOD_DATA
T.Phys-Probing	Physical Probing	An attacker may perform physical probing of the TOE in order: (i) to disclose User Data (ii) to disclose/reconstruct the Security IC Embedded Software or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.	T.DIS_ES2 T.MOD_TSF T.MOD_SOFT* T.DIS_DATA
T.Malfunction	Malfunction due to Environmental Stress	An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions.	T.DIS_ES2 T.T_CMD T.MOD_TSF T.MOD_EXE T.LOAD_MAN T.LOAD_APP T.APP_CORR T.APP_REMOVE T.ERR_REMOVE T.DEL_REMOVE T.APP_READ T.APP_MOD T.DIS_DATA T.MOD_DATA
T.Leak-Inherent	Inherent Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets. No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements.	T.DIS_ES2 T.DIS_DATA
T.Leak-Forced	Forced Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets even if the information leakage is not inherent but caused by the attacker.	T.DIS_ES2 T.DIS_DATA
T.Abuse-Func	Abuse of Functionality	An attacker may use functions of the TOE which may not be used after TOE Delivery (e.g. test features) in order to: (i) disclose or manipulate User Data (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software.	T.DIS_ES2 T.MOD_SOFT* T.APP_CORR T.APP_REMOVE T.ERR_REMOVE T.DEL_REMOVE T.APP_READ T.APP_MOD T.DIS_DATA T.MOD_DATA
T.RND	Deficiency of Random Numbers	An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.	
T.Masquerade_TOE	Masquerade of the TOE	An attacker may threaten the property being a genuine TOE by producing a chip which is not a genuine TOE but wrongly identifying itself as genuine TOE sample.	T.DIS_INFO* TT_DEL
T.Mem-Access	Memory Access Violation	Parts of the Smartcard Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code) or privilege levels. Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard Embedded Software	T.MOD_SOFT* T.T_CMD T.MOD_LOAD, T.MOD_EXE, T.MOD_SHARE T.APP_READ T.APP_MOD

IDMotion V2 Virtual Machine Security Target

IC Relevant Threat Label	IC Relevant Threat Title	IC Relevant Threat Content	Link to the composite-product threats
T.Open_Samples_Diffusion	Diffusion of open samples	An attacker may get access to open samples of the TOE and use them to gain information about the TSF (loader, memory management unit, ROM code ...). He may also use the open samples to characterize the behavior of the IC and its security functionalities (for example: characterization of side channel profiles, perturbation cartography ...). The execution of a dedicated security features (for example: execution of a DES computation without countermeasures or by de-activating countermeasures) through the loading of an adequate code would allow this kind of characterization and the execution of enhanced attacks on the IC.	T.DIS_INFO* T.DIS_DEL* T.DIS_DEL1 T.DIS_DEL2 T.T_SAMPLE2 T.MOD_DEL* T.MOD_DEL1 T.MOD_DEL2 T.T_ES

Note. Both T.DIS_ES1 and T.CLON* relate to Phase 1 which is before the existence of the IC product and T.RND is irrelevant to the composite product. Therefore, for this evaluation, all three of these threats are considered outside the scope of the TOE.

4.5.2 Statement of Compatibility – OSPs

The following table lists the relevant OSPs of the IFX_CCI_000014 and provides the link to the OSPs related to the composite product, showing that there is no contradiction between the two.

IC OSP Label	IC OSP Content	Link to the composite-product
P.Process-TOE	<p>Identification during TOE Development and Production</p> <p>An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.</p>	No contradiction with the present evaluation OSPs. Current evaluation has objectives related to delivery to IC manufacturer such as O.DLV_DATA, O.SOFT_DLV* and O.DLV_PROTECT*
P.Add-Functions	<p>Additional Specific Security Functionality</p> <p>The TOE shall provide the following specific security functionality to the Smartcard Embedded Software:</p> <p>Rivest-Shamir-Adleman Cryptography (RSA)</p> <p>Elliptic Curve Cryptography (EC)</p>	No contradiction with the present evaluation. Present evaluation makes use of RSA and ECC functions.
P.Crypto- Service	<p>Cryptographic services of the TOE</p> <p>The TOE provides secure hardware based cryptographic services for the IC Embedded Software:</p> <p>Triple Data Encryption Standard (TDES)</p> <p>Advanced Encryption Standard (AES)</p>	No contradiction with the present evaluation. Present evaluation makes use of DES and AES functions.
P.Lim_Block_Loader	<p>Limiting and Blocking the Loader Functionality</p> <p>The composite manufacturer uses the Loader for loading of Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.</p>	No contradiction with the present evaluation OSPs. The loader is erased by the OS Image during the Phase 5 of the TOE life cycle

IDMotion V2 Virtual Machine Security Target

P.Ctrl_Loader	<p>Controlled usage to Loader Functionality Authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation.</p>	No contradiction with the present evaluation OSPs. Current evaluation has objectives related to user control such as O.DLV_DATA
---------------	---	---

4.5.3 Statement of Compatibility – Assumptions

The following table lists the relevant assumptions of the IFX_CCI_000014 and provides the link to the assumptions related to the composite product, showing that there is no contradiction between the two.

IC assumption label	IC assumption title	IC assumption content	IrPA	CfPA	SgPA	Link to the composite product
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalization	It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).			X	<p>A.DLV_PROTECT* A.DLV_AUDIT* A.DLV_RESP* A.USE-TEST* A.USE_PROD* A.USE-DIAG*</p> <p>The assumptions are the same.</p>
A.Resp-Appl	Treatment of User Data	All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.		X		<p>Assets have to be protected in terms of confidentiality, authenticity and control of their origin. Assets linked to:</p> <ul style="list-style-type: none"> • IC and MAP specific data • MULTOS Initialization Security Data • MSM Controls Data • NA TSF data := keys and identification data • Users of Loaded Applications • Loaded Application software • Loaded Application SF data <p>The OS owns security relevant User Data.</p> <p>O.DIS_MEMORY* O.MOD_MEMORY* O.LOAD O.REMOVE O.SECURITY O.SEGREGATE</p>

IDMotion V2 Virtual Machine Security Target

IC assumption label	IC assumption title	IC assumption content	IrPA	CfPA	SgPA	Link to the composite product
A.Key-Function	Usage of Key-dependent Functions	<p>Key-dependent functions, if any, shall be implemented in the Security IC Embedded Software in a way that they are not susceptible to leakage attacks (as described under BSI.T.Leak-Inherent and BSI.T.Leak-Forced).</p> <p>Note that here the routines which may compromise keys when being executed are part of the Smartcard Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE.</p>		X		<p>O.TAMPER_ES O.SIDE O.DEV_DIS_ES O.INIT_ACS</p>

4.5.4 Statement of Compatibility – Security Objectives

The following table lists the relevant security objectives of the IFX_CCI_000014 and provides the link to the security objectives related to the composite product, showing that there is no contradiction between the two.

IC objective label	IC objective title	Link to the composite-product
O.Phys-Manipulation	Protection against Physical Manipulation	<p>O.TAMPER_ES O.SIDE O.CLON* O.FLAW*</p> <p>O.DIS_MECHANISM2 O.DIS_MEMORY*</p>
O.Phys-Probing	Protection against Physical Probing	<p>O.TAMPER_ES O.SIDE O.CLON*</p>
O.Malfunction	Protection against Malfunction due to Environmental Stress	<p>O.TAMPER_ES O.SIDE</p>
O.Leak-Inherent	Protection against Inherent Information Leakage	<p>O.MOD_MEMORY* O.SIDE</p>
O.Leak-Forced	Protection against Forced Information Leakage	<p>O.DIS_MEMORY* O.SIDE</p>
O.Abuse-Func	Protection against Abuse of Functionality	<p>O.TAMPER_ES O.SIDE O.OPERATE* O.DIS_MECHANISM2</p>
O.Identification	TOE Identification	No contradiction with the present evaluation.
O.RND	Random Numbers	No contradiction with the present evaluation.
O.Cap_Avail_Loader	Capability and availability of the Loader Valid only for the TOE derivatives delivered with activated Flash Loader.	<p>O.MOD_MEMORY* O.ROLLBACK O.MOD_MEMORY* O.SAMPLE_ACS O.TEST_OPERATE*</p>

IDMotion V2 Virtual Machine Security Target

IC objective label	IC objective title	Link to the composite-product
O.Authentication	Authentication to external entities Valid only for the TOE derivatives delivered with activated Flash Loader	O.RESOURCE O.SAMPLE_ACS O.INIT_A
O.Ctrl_Auth_Loader	Access control and authenticity for the Loader - valid only for the TOE derivatives delivered with activated Flash Loader	O.DEV_DIS_ES O.SOFT_DLTV* O.INIT_A
O.TDES	Cryptographic service Triple-DES	No contradiction with the present evaluation. Present evaluation makes use of DES
O.AES	Cryptographic service AES	No contradiction with the present evaluation. Present evaluation makes use of AES
O.Add-Functions	Additional Specific Security Functionality The TOE must provide the following specific security functionality to the Smartcard Embedded Software: Rivest-Shamir-Adleman Cryptography (RSA) Elliptic Curve Cryptography (EC)	No contradiction with the present evaluation. Present evaluation makes use of RSA and ECC
O.Mem Access	Area based Memory Access Control The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that access of software to memory areas and privilege levels is controlled as required, for example, in a multi-application environment.	O.DIS_MECHANISM2 O.DIS_MEMORY* O.MOD_MEMORY* O.RESOURCE O.LOAD O.SECURITY O.EFFECT_L O.REMOVE O.EFFECT_R O.SEGREGATE
O.Prot_TSF_Confidentiality	Protection of the confidentiality of the TSF The TOE must provide protection against disclosure of confidential operations of the Security IC (loader, memory management unit ...) through the use of a dedicated code loaded on open samples.	O.SAMPLE_ACS O.INIT_ACS O.DEV_DIS_ES O.DEV_TOOLS* O.MOD_MEMORY* O.DIS_MEMORY* O.DIS_MECHANISM2 O.FLAW* O.OPERATE*
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader The Composite Product Manufacturer will protect the Loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader.	No contradiction with the present evaluation as the loader tool is overwrite during the phase 5 of the TOE life cycle
OE.TOE_Auth	Authentication to external entities The operational environment shall support the authentication verification mechanism and know authentication reference data of the TOE.	O.DEV_TOOLS* O.DEV_DIS_ES O.INIT_ACS O.DLV_PROTECT* O.APPLI_DEV
OE.Loader_Usage	Secure communication and usage of the Loader The authorized user must support the trusted communication with the TOE by confidentiality protection and authenticity proof of the data to be loaded and fulfilling the access conditions required by the Loader.	O.DEV_DIS_ES O.SOFT_DLTV* O.INIT_ACS O.SAMPLE_ACS O.DLV_PROTECT* O.DLV_AUDIT* O.DLV_RESP* O.TEST_OPERATE*

IDMotion V2 Virtual Machine Security Target

4.5.5 Statement of Compatibility – SFRs

The following table lists the relevant SFRs of the IFX_CCI_000014 and provides the link to the SFRs related to the composite product, showing that there is no contradiction between the two.

Security Functional Requirement		IP_SFR	RP_SFR	Composite product SFRs
FAU_SAS.1	Audit storage		X	FMT_MSA.1
FCS_CKM.4/AES	Cryptographic key destruction – AES		X	No link
FCS_CKM.4/TDES	Cryptographic key destruction – TDES		X	
FCS_COP.1/AES	Cryptographic operation – AES		X	No link
FCS_COP.1/TDES	Cryptographic operation – TDES		X	
FCS_RNG.1/DRNG	Random number generation - DRNG		X	No link
FCS_RNG.1/HPRG	Random number generation – HPRG		X	No link
FCS_RNG.1/KSG	Random number generation - KSG		X	No link
FCS_RNG.1/TRNG	Random number generation - TRNG		X	No link
FDP_ACC.1/Loader	Subset access control – Loader		X	No direct link to the composite product SFRs, since the IC Loader is no more available after phase 5. However, these two IC TOE SFRs are essential to protect the composite TOE during phases 4 and 5 (covered by the ALC assurance classes)
FDP_ACF.1/Loader	Security attribute based access control – Loader		X	
FDP_IFC.1	Subset information flow control		X	No link
FDP_ITT.1	Basic internal transfer protection		X	FAU_SAA.1.1 & FAU_SAA.1.2 FAU_ARP.1.1
FDP_SDC.1	Stored data confidentiality		X	FPT_FLS.1.1 FPT_FLS.1
FDP_SDI.2	Stored data integrity monitoring and action		X	FAU_ARP.1.1 FAU_SAA.1.1 & FAU_SAA.1.2
FDP_UCT.1	Basic data exchange confidentiality		X	No direct link to the composite product SFRs, since the IC Loader is no more available after phase 5
FDP_UIT.1	Data exchange integrity		X	
FIA_API.1	Identification and Authentication			FMT_MSA.1
FMT_LIM.1	Limited capabilities		X	FDP_ACC.2 FDP_ACF.1
FMT_LIM.2	Limited availability		X	
FMT_LIM.1/Loader	Limited capabilities		X	No direct link to the composite product SFRs, since the IC Loader is no more available after phase 5
FMT_LIM.2/Loader	Limited availability		X	
FPT_FLS.1	Failure with preservation of secure state		X	FAU_ARP.1.1 FAU_SAA.1.1 FPT_FLS.1
FPT_ITT.1	Basic internal TSF data transfer protection		X	No link

IDMotion V2 Virtual Machine Security Target

FPT_PHP.3	Resistance to physical attack		X	No link
FRU_FLT.2	Limited fault tolerance		X	FAU_ARP.1.1 FAU_SAA.1.1 FPT_FLS.1
FTP_ITC.1	Inter-TSF Trusted Channel			FDP_ACC.2 FDP_ACF.1
FPT_TST.2	Subset TOE security testing		X	No link
FDP_ACC.1	Subset access control		X	FDP_ACC
FDP_ACF.1	Security attribute based access control		X	FDP_ACF.1
FMT_MSA.1	Management of security attributes		X	No link
FMT_MSA.3	Static attribute initialization		X	
FMT_SMF.1	Specification of Management functions		X	FDP_ACC.2 FDP_ACF.1 FPT_FLS
FDP_SDI.1	Stored data integrity monitoring		X	No link
FCS_COP.1/RSA	Cryptographic Operation – RSA		X	No link
FCS_CKM.1/RSA	Cryptographic key management - RSA		X	No link
FCS_COP.1/ECDSA	Cryptographic Operation – ECDSA		X	No link
FCS_CKM.1/EC	Cryptographic key management - EC		X	No link
FCS_COP.1/ECDH	Cryptographic Operation – ECDH		X	No link

5. SECURITY OBJECTIVES

5.1 SECURITY OBJECTIVES FOR THE TARGET OF EVALUATION

The TOE shall achieve the following IT security objectives.

O.FLAW*

The TOE must not contain flaws in design, implementation or operation.

O.ROLLBACK

The TOE must be in a well-defined valid state before the loading of an application, even in case of failure of the previous loading or removal. A failure must not hinder the resources that the TOE can deliver. A rollback operation can be achieved either through specific commands or automatically.

O.RESOURCE

The TOE must provide the means of controlling the use of resources by its users and subjects so as to prevent permanent unauthorized denial of service. (For example it must prevent a Loaded-Application from taking control of the whole permanent memory (EEPROM) thus prohibiting other Loaded-Applications from using it).

O.EFFECT_LOAD

Loading an application must have no effect on the code and data of existing Loaded-Applications.

O.EFFECT_REMOVE

Removal of a Loaded-Application must have no effect on the code and data of the remaining independent Loaded-Applications.

O.SEGREGATE

Loaded-Applications are to be segregated from other Loaded-Applications. A Loaded-Application may not read from or write to another Loaded-Application's code or data without its authorization.

5.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

5.2.1 Objectives on Phase 1

Note that these objectives for phase 1 are described to maintain compatibility with PP/0010 which is compliant to Common Criteria v2.1. Common Criteria v3 does not require objectives for the development environment so these objectives (and any references to them) should be ignored for this version of Common Criteria. Instead, Common Criteria v3 requires that the development environment is evaluated in the ALC assurance class of the evaluation.

O.DEV_TOOLS*

The Smartcard ES shall be designed in a secure manner, by using exclusively software development tools (compilers, assemblers, linkers, simulators, etc.) and software-hardware integration testing tools (emulators) that will result in the integrity of program and data.

O.DEV_DIS_ES

The Embedded Software developer shall use established procedures to control storage and usage of the classified development tools and documentation, suitable to maintain the integrity and the confidentiality of the assets of the TOE.

It must be ensured that tools are only delivered and accessible to the parties authorized personnel.

It must be ensured that confidential information on defined assets is only delivered to the parties' authorized personnel on a need-to-know basis.

O.SOFT_DL*

The Embedded Software must be delivered from the Smartcard software developer (Phase I) to the IC designer through a trusted delivery and verification procedure that shall be able to maintain the integrity of the software and its confidentiality, *if applicable*.

NOTE: In PP00/10 it will be always considered applicable.

O.INIT_ACS

Initialization Data shall be accessible only by authorized personnel (physical, personnel, organizational, technical procedures).

O.SAMPLE_ACS

Samples used to run tests shall be accessible only by authorized personnel.

5.2.2 Objectives on the Target of Evaluation Delivery Process (Phases 4 to 7)

O.DLV_PROTECT*

Procedures shall ensure protection of TOE material/information under delivery including the following objectives:

- Non-disclosure of any security relevant information.
- Identification of the element under delivery.
- Meet confidentiality rules (confidentiality level, transmittal form, reception acknowledgement).
- Physical protection to prevent external damage.
- Secure storage and handling procedures (including rejected TOEs).
- Traceability of TOE during delivery including the following parameters:
 - Origin and shipment details.
 - Reception, reception acknowledgement.
 - Location material/information.

O.DLV_AUDIT*

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process (including if applicable any non-conformance to the confidentiality convention) and highlight all non-conformance to this process.

O.DLV_RESP*

Procedures shall ensure that people (shipping department, carrier, reception department) dealing with the procedure for delivery have got the required skill, training and knowledge to meet the procedure requirements and be able to act fully in accordance with the above expectations.

5.2.3 Objectives on Delivery from Phase 1 to Phases 4, 5 and 6

O.DLV_DATA

Native-Application and ES data must be delivered from the Smartcard embedded software developer (phase 1) either to the IC Packaging manufacturer, the Finishing Process manufacturer or the Personaliser through a trusted delivery and verification procedure that shall be able to maintain the integrity and confidentiality of the ES (Note: some application data are not required for embedding and are then delivered directly to phases 4 to 6).

5.2.4 Objectives on Phases 4 to 6

O.TEST_OPERATE*

Appropriate functionality testing of the TOE shall be used in phases 4 to 6.

During all manufacturing and test operations, security procedures shall be used through phases 4, 5 and 6 to maintain confidentiality and integrity of the TOE and its manufacturing and test data.

5.2.5 Objectives on Phase 7

O.USE_DIAG*

Secure communication protocols and procedures shall be used between the Smartcard and the terminal.

5.2.6 Additional objectives for the operational environment

The following objectives are not ensured by the TOE (i.e. the Application Abstract Machine and the Application Memory Manager) but by the other components of the MULTOS OS.

O.TAMPER_ES

The TOE must prevent tampering with its security critical parts. In particular, the security mechanisms must prevent the unauthorized change of functional parameters, security attributes and secrets such as the life cycle sequence flags and cryptographic keys.

O.SIDE

The ES must be designed to avoid interpretations of electrical signals from the hardware part of the TOE.

O.CLON*

The TOE functionality must be protected from cloning.

O.OPERATE*

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

O.DIS_MEMORY*

The TOE shall ensure that sensitive information stored in memories is protected against unauthorized disclosure.

NOTE sensitive information means User Data and TSF data.

O.MOD_MEMORY*

The TOE shall ensure that *sensitive information* stored in memories is protected against any corruption or unauthorized modification.

NOTE sensitive information means User Data and TSF data.

The following security objectives are necessary to meet the new threats specific to Multi-Application Platforms. This is why these objectives are new and not present in PP/9911.

O.DIS_MECHANISM2

The TOE shall ensure that the ES security mechanisms are protected against unauthorized disclosure.

O.LOAD

Loaded-Applications are only to be loaded onto a platform with the permission of the administrator.

O.SECURITY

The application load process must be able to guarantee, when required, the integrity, confidentiality, and to verify the claimed origin of the Loaded-Application code and data.

O.REMOVE

Removal of a Loaded-Application and consequent reuse of the Loaded-Application space is only to be performed with the authorization of the administrator. The space must not hold any information relative to data or code linked to the removed Loaded-Application.

O.CIPHER

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

O.DECIPHER

The TOE shall provide a means to decipher the KTUs.

5.2.7 Objectives on Loaded-Application Development and Loading (Phases A1 and A2)

This Objective is specific to Loaded-Application development in the Smartcard IC with Multi-Application Platform environment.

O.APPLI_DEV

The Loaded-Application provider must:

- Follow the Administrator Guidance and ensure that the applications are compliant with the Security Guidance for Application Developers. Amongst other topics application should ensure that sensitive assets are suitably protected (ie encrypted)
- Provide trusted delivery channel so that the integrity and origin of the Loaded-Application can be verified and that its confidentiality can be maintained.

5.3 SECURITY OBJECTIVES RATIONALE

This section demonstrates that the stated specific security objectives address, and can be traced to, all security environment aspects identified: each specific security objective being correlated to at least one threat, one OSP or one assumption.

5.3.1 Discussion of Threats, OSP and Security Objectives

The following discussion shows which security objectives counter which threats and enforce which OSP, phase by phase.

During phase 1, the Smartcard ES is being developed and the pre-personalization and personalization requirements are specified for all other phases. The Target of Evaluation (TOE) is a functional product designed during phase 1, considering that the only purpose of the Embedded Software is to control and protect the operation of the Smartcard during phase 4 to 7 (operational phases). The global security requirements of the TOE mandate to consider, during the development phase, the security threats of the other phases. This is why the PP addresses the functions used in phases 4 to 7 but developed during phase 1. Then, the limit of the TOE corresponds to the phase 1 including the TOE delivery to the IC manufacturer.

T.CLON*

The TOE being constructed can be cloned, but also the construction tools and document can help clone it. During phase 1, since the product does not exist, it cannot contribute to countering the threat. For the remaining phases 4 to 7, TOE participates in countering the threats.

T.DIS_INFO*

This threat addresses disclosure of specification, design and development tools concerning the IC and delivered to the software developer (during phase 1) in order to meet with the overall security objectives of the TOE. This threat is countered by development environment.

T.DIS_DEL*

This threat addresses disclosure of specifications, test programs, related documents, ES and data which is delivered from phase 1 to phase 2 for software embedding. As the TOE does not yet exist, the threat can only be countered by development environmental procedures.

T.DIS_DEL1

This threat addresses disclosure of software and ES Data during delivery from phase 1 to phases 4 to 6. As the data is not yet implemented in the TOE, the threat can only be countered by environmental procedures.

T.DIS_DEL2

This threat addresses disclosure of software or data which has been delivered, from phase 1, to phases 4 to 6. As the data is not yet implemented in the TOE, the threat can only be countered by environmental procedures.

T.DIS_ES1

The ES and accompanying documents are created and used during phase 1. As during this phase the product does not yet exist, it cannot contribute to countering the threat which must be countered by development environment.

T.DIS_ES2

Disclosure of ES and TSF data can compromise security. During phases 4 to 7, the TOE must counter the unauthorized disclosure of the ES and the Loaded-Application Data.

T.DIS_TEST_ES

Tests concerning the embedded software or software to be embedded are carried out in phase 1. This threat is countered by environmental development procedures, of which the tests themselves are part.

TT_DEL

The threat addresses the theft of software or ES Data which is delivered for software embedding, from phase 1 to phase 2. As the data is not yet implemented in the TOE, the threat can only be countered by developmental environmental procedures.

T.T_TOOLS

TOE development tools are used only during phase 1, so this threat can only exist during phase 1. As the TOE is not yet manufactured, this threat is countered by environmental procedures.

T.T_SAMPLE2

TOE samples are used only during phase 1, so this threat can only exist during phase 1. The theft or unauthorized use of samples are countered by environmental procedures.

T.MOD_DEL*

This threat addresses modification of software or TSF data which is delivered for software embedding, in phase 1. As the TOE does not exist during this phase, the threat must be countered by development procedures.

T.MOD_DEL1

This threat addresses modification of ES Personalization Data during delivery from embedded software developer, phase 1, to the IC packaging manufacturer, phase 4, the finishing process manufacturer, phase 5, and for the Personalizer, phase 6. As the data is not yet loaded on the TOE, the threat can only be countered by environmental procedures.

T.MOD_DEL2

This threat addresses modification of ES Personalization Data which is delivered to the IC packaging manufacturer, phase 4, the finishing process manufacturer, phase 5, and for the Personalizer, phase 6. As the data is not yet loaded on the TOE, the threat can only be countered by environmental procedures.

T.MOD

Modification of ES and TSF Data can be done during ES design in phase 1. Since the product does not exist, the threat can only be countered by environment procedures.

T.MOD_SOFT*

Once present on the TOE, the software and Application data can be modified in an unauthorized way during any phases from 4 to 7. This threat is countered by the TOE.

T.T_ES

This threat covers the unauthorized use of cards during the different phases of the card life cycle as well as the misappropriation of rights of Smartcards. This threat covers phases 4 to 7 and is countered by the TOE.

T.T_CMD

This threat includes the diversion of the hardware or the software, or both, in order to execute non authorized operations. This threat covers phases 4 to 7 and is countered by the TOE.

T.MOD_LOAD, T.MOD_EXE, T.MOD_SHARE

The loading of Native Applications, execution and modification of software can endanger the security of the TOE, and especially create interference between applications. This threat covers phases 4 to 7 and is countered by the TOE.

New threats not specific to Multi-Application platforms:

T.MOD_TSF

Modification of TOE Security function can only appear when the TOE exists, thus only during phases 4 to 7. This threat is countered by the TOE.

T.DIS_DATA

Threats on end user data can only appear after the data has been created, thus in usage phase 7. This threat is countered by the TOE.

T.MOD_DATA

Threats on end user data can only appear after the data has been created, thus in usage phase 7. This threat is countered by the TOE.

New Threats specific to Multi-Application platforms:

These threats are present during phases 6 to 7 depending when the Loaded-Applications are loaded. It can be supposed that Loaded-Applications are mostly used during phase 7.

T.LOAD_MAN

This threat comes from illegal loading of Loaded-Applications which can for example clone legal Loaded-Applications on other cards. Loading can be done in phases 6 and 7. This threat is countered by the TOE.

T.LOAD_APP

This threat is a complement to the precedent one. In this case, an illegal Loaded-Application is loaded in place of a legal one. The attacking party can be the same as above. This threat appears during phases 6 and 7 and is countered by the TOE.

T.LOAD_OTHER

This threat addresses loading a Loaded-Application to a domain to which it should not have access. This means that the other Loaded-Application can be attacked. This threat appears during phases 6 and 7 and is countered by the TOE.

T.LOAD_MOD

This threat alters code or data without the permission of the Loaded-Application Provider. This threat appears during phases 6 and 7 and is countered by the TOE and the environment.

T.APP_DISC

This is an attack on the Loaded-Application provider know-how, and possibly on confidential data loaded along with the Loaded-Application. This threat appears during phases 6 and 7 and is countered by the TOE and the environment.

T.APP_CORR

This attack destroys partly or completely the other Loaded-Application, or more subtly can divert the Loaded-Application to create a dangerous state. This threat appears during phases 6 and 7 and is countered by the TOE.

T.APP_REMOVE

This threat addresses illegal removal of a legal Loaded-Application. It attacks reliability of services. This threat appears during phases 6 and 7 and is countered by the TOE.

T.ERR_REMOVE

This opportunistic threat takes advantage of a removal operation to attack the confidentiality of Loaded-Application provider know how, or confidential data. This threat appears during phases 6 and 7 and is countered by the TOE.

T.DEL_REMOVE

This threat is on remaining Loaded-Applications which can be damaged during the removal operation. This threat appears during phases 6 and 7 and is countered by the TOE.

T.APP_READ

This threat loads a Trojan horse to illegally access to confidential data belonging to other Loaded-Applications. This threat appears during phases 6 and 7 and is countered by the TOE.

T.APP_MOD

This threat loads a Trojan horse to illegally access to modify data or code belonging to another Loaded-Application. This threat appears during phases 6 and 7 and is countered by the TOE.

T.RESOURCES

This threat is aimed at the reliability of service of the platform or Loaded-Application. This threat appears during phases 6 and 7 and is countered by the TOE.

Threat T.APP_DISC is also present during Loaded-Application development phase A1 when it is countered by the environment.

5.3.2 Threats & OSP Addressed by Security Objectives

5.3.2.1 Security objectives for the TOE

Threats/OSP/Obj	O.FLAW*
T.DIS.ES2	X
T.T_ES	X
T.T_CMD	X
T.MOD_SOFT*	X
T.MOD_LOAD	X
T.MOD_EXE	X
T.MOD_SHARE	X
T.MOD_TSF	X
T.DIS_DATA	X
T.MOD_DATA	X

Table 6: Mapping of security objectives to threats and OSP relative to phases 4 to 7

T.DIS_ES2

Illegal disclosure can be achieved by incorrect operation of the TOE, which is countered by O.FLAW*.

T.T_ES

Unauthorized use of TOE can be achieved by a degradation of the security mechanisms which is countered by O.FLAW*.

T.T_CMD

To be able to have an unauthorized use of sequences sent to the TOE, it is necessary to achieve a degradation of the security mechanisms which is countered by O.FLAW*.

T.MOD_SOFT*

The modification of embedded software of the TOE is countered by a correct operation of security mechanisms O.FLAW*.

T.MOD_EXE

IDMotion V2 Virtual Machine Security Target

To be able to illegally execute programs on TOE, it is necessary to bypass or degrade the access security mechanisms. This is countered by O.FLAW*.

T.MOD_LOAD

To be able to load illegally programs on TOE, it is necessary to achieve a degradation of the security mechanisms which is countered by O.FLAW*.

T.MOD_SHARE

To be able to modify programs on TOE, it is necessary to bypass or degrade security mechanisms. This is countered by O.FLAW*.

T.MOD_TSF

Absence of design flaws, O.FLAW*, is necessary to counter the threat.

T.DIS_DATA

Absence of design flaws, O.FLAW*, is necessary to counter the threat.

T.MOD_DATA

Absence of design flaws, O.FLAW*, is necessary to counter the threat.

Threats/OSP/Obj	O.ROLLBACK	O.RESOURCE	O.EFFECT_LOAD	O.EFFECT_REMOVE	O.SEGREGATE
T.LOAD_OTHER			X		
T.APP_CORR			X		
T.DEL_REMOVE				X	
T.APP_READ			X		
T.APP_MOD			X		
T.RESOURCES	X	X			

Table 7: Mapping of security objectives to threats & OSP relative to phases 6 and 7

The TOE shall use state of the art technology to achieve the following IT security objectives and enforce the OSP.

T.LOAD_OTHER

Loading an application into another illegal domain is countered by O.EFFECT_LOAD, which prevents applications from having non-authorized effects on applications loaded in other domains.

T.APP_CORR

Loading an application so it corrupts another application is countered by O.EFFECT_LOAD, which prevents applications from having non-authorized effects on applications loaded in other domains.

T.DEL_REMOVE

Deletion of part of a Loaded Application by removal of another is countered by O.EFFECT_REMOVE, which ensures that removal has no effect on other Loaded Applications.

T.APP_READ

Use of a Loaded Application to illegally read data contained in another application is countered by O.SEGREGATE which ensure that illegal reading of data of another application is not possible.

T.APP_MOD

IDMotion V2 Virtual Machine Security Target

Use of a Loaded Application to illegally modify data or code contained in another application is countered by O.SEGREGATE, which ensure that illegal modification of data or code of another application is not possible.

T.RESOURCES

Destruction or hoarding of card resources is prevented by O.ROLLBACK which guarantees that a failure does not compromise card resources and by O.RESOURCES which controls the use of card resources by Loaded Applications.

5.3.2.2 Security objectives for the environment

The following descriptions and tables map the security objectives for the environment relative to the various threats countered, assumptions upheld and OSP enforced, in addition to the Smartcard PP.

Threats/OSP/Obj	O.TAMPER_ES	O.SIDE	O.OPERATE*	O.DIS_MECHANISM2	O.DIS_MEMORY*	O.MOD_MEMORY*	O.CLON*
T.CLON*				X	X		X
T.DIS_ES2		X	X	X	X		
T.T_ES	X		X			X	
T.T_CMD	X		X			X	
T.MOD_SOFT*	X		X			X	
T.MOD_LOAD	X		X			X	
T.MOD_EXE	X		X		X	X	
T.MOD_SHARE	X		X		X	X	
T.MOD_TSF	X		X			X	
T.DIS_DATA		X	X		X		
T.MOD_DATA	X		X			X	

Table 8: Mapping of security objectives for the environment to threats, assumptions and OSP relative to phases 4 to 7

The TOE environment shall use state of the art technology to achieve the following IT security objectives and enforce the OSP; for that purpose, when IC physical security features are used, the specification of these physical security features shall be respected:

T.CLON*

The general threat is countered by the dedicated objective O.CLON*.

T.DIS_ES2

Illegal disclosure of ES is countered by O.DIS_MECHANISM2 and disclosure of Application by O.DIS_MEMORY*. More specifically this can be achieved by incorrect operation of the TOE, which is countered by O.OPERATE*, or by a direct observation during operation which is countered by O.SIDE

T.T_ES

Unauthorized use of TOE can be achieved by a degradation of the security mechanisms which is countered by O.OPERATE*, or by modification of the security mechanisms countered by O.TAMPER_ES, or by modification of TSF data, countered by O.MOD_MEMORY*.

T.T_CMD

IDMotion V2 Virtual Machine Security Target

To be able to have an unauthorized use of sequences sent to the TOE, it is necessary to achieve a degradation of the security mechanisms which is countered by O.OPERATE*, or to modify the security mechanisms countered by O.TAMPER_ES or by modification of TSF data, countered by O.MOD_MEMORY*.

T.MOD_SOFT*

The modification of embedded software of the TOE is countered by a correct operation of security mechanisms O.OPERATE*. The threat includes modification of the security mechanisms themselves which is countered by O.TAMPER_ES and modification of TSF data, countered by O.MOD_MEMORY*.

T.MOD_EXE

To be able to illegally execute programs on TOE, it is necessary to bypass or degrade the access security mechanisms. This is countered by O.OPERATE*. Modification of the security mechanisms is countered by O.TAMPER_ES. . It is also possible to gain access through modification of TSF data, which is countered by O.MOD_MEMORY*, or through disclosure of TSF data which is countered by O.DIS_MEMORY*.

T.MOD_LOAD

To be able to load illegally programs on TOE, it is necessary to achieve a degradation of the security mechanisms which is countered by O.OPERATE*, or to modify the security mechanisms countered by O.TAMPER_ES or by modification of TSF data, countered by O.MOD_MEMORY*.

T.MOD_SHARE

To be able to modify programs on TOE, it is necessary to bypass or degrade security mechanisms. This is countered by O.OPERATE*. Modification of the security mechanisms is countered by O.TAMPER_ES. Illegal Modification of Application data is countered by O.MOD_MEMORY*. This is also countered by protection of the confidentiality of TSF data: O.DIS_MEMORY*.

T.MOD_TSF

The illegal modification of TSF data of the TOE is countered by O.MOD_MEMORY* which addresses also TSF data. It is also possible to degrade or bypass access mechanisms, which is countered by O.TAMPER_ES and O.OPERATE*.

T.DIS_DATA

The disclosure of application user and TSF data on the TOE is countered by O.DIS_MEMORY*. To fulfill the threat, it is necessary to degrade or bypass access mechanisms, which is countered by O.SIDE and O.OPERATE*.

T.MOD_DATA

The modification of application user data on the TOE is countered by O.MOD_MEMORY*. To fulfill the threat, it can be necessary to degrade or bypass access mechanisms, which is countered by O.TAMPER_ES, O.OPERATE*.

ThreatsAssumptions /OSP/Obj	O.DEV_TOOLS*	O.DEV_DIS_ES	O.SOFT_DLV*	O.INIT_ACS	O.SAMPLE_ACS	O.DLV_PROTECT	O.DLV_AUDIT	O.DLV_RESP*	O.TEST_OPERATE*	O.USE_DIAG*	O.APPLI_DEV
T.CLON*		X	X	X	X						
T.DIS_INFO*		X									
T.DIS_DEL*			X								
T.DIS_ES1		X		X							
T.DIS_TEST_ES		X									
T.T_DEL*			X								
T.T_TOOLS	X										
T.T_SAMPLE2					X						
T.MOD_DEL*			X								

IDMotion V2 Virtual Machine Security Target

Threats/Assumptions /OSP/Obj	O.DEV_TOOLS*	O.DEV_DIS_ES	O.SOFT_DLTV*	O.INIT_ACS	O.SAMPLE_ACS	O.DLV_PROTECT	O.DLV_AUDIT	O.DLV_RESP*	O.TEST_OPERATE*	O.USE_DIAG*	O.APPLI_DEV
T.MOD		X		X							
A.DLV_PROTECT*						X					
A.DLV_AUDIT*							X				
A.DLV_RESP*								X			
A.USE_TEST*									X		
A.USE_PROD*									X		
A.USE_DIAG*										X	
A.APPLI_CONT											X

Table 9 : Mapping of security objectives for the environment to threats, assumptions and OSP relative to phase 1

T.CLON*

Cloning requires knowledge of:

- Development data and access to tools, which is countered by O.DEV_DIS_ES
- The software which is countered by O.SOFT_DLTV*
- Initialization data which is countered by O.INIT_ACS

Cloning can also be done by using samples; this is countered by O.SAMPLES_ACS.

T.DIS_INFO*

Disclosure of IC assets is countered by O.DEV_DIS_ES, which guarantees the storage of classified information.

T.DIS_DEL*

Disclosure of embedded software and corresponding data during delivery is countered by O.SOFT_DLTV*.

T.DIS_ES1

Disclosure of ES is countered by O.DEV_DIS_ES, which guarantees the storage of classified information, and by O.INIT_ACS which guarantees a controlled access to initialization data.

T.DIS_TEST_ES

Disclosure of ES test program is countered by O.DEV_DIS_ES, which guarantees the storage of classified information.

T.T_EL*

Theft of software delivered to IC manufacturer is countered by O.SOFT_DLTV* which ensures trusted delivery.

T.T_TOOLS

Theft or unauthorized access to development tools is countered by O.DEV_TOOLS* which controls the accesses.

T.T_SAMPLE2

Theft of samples is countered by O.SAMPLE_ACS controlled access.

T.MOD_DEL*

Modification of software and related information is countered O.SOFT_DLTV*.

T.MOD

Unauthorized modifications of software are countered by access control specified by O.DEV_DIS_ES and that of TSF data by O.INIT_ACS.

IDMotion V2 Virtual Machine Security Target

A.DLV_PROTECT*

Procedures shall ensure protection of TOE material/information under delivery and storage.

A.DLV_AUDIT*

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage.

A.DLV_RESP*

Procedures shall ensure that people dealing with the procedure for delivery have got the required skill.

A.USE_TEST*

It is assumed that appropriate functionality testing of the TOE is used in phases 4, 5 and 6.

A.USE_PROD*

It is assumed that security procedures are used during all manufacturing and test operations through phases 4, 5, 6 to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

A.USE_DIAG*

It is assumed that secure communication protocols and procedures are used between Smartcard and terminal.

A.APPLI_CONT

Whenever a Loaded-Application is to be loaded on the platform, it is assumed that its development and production follow the Administrator Guidance.

Threats	O.DLV_DATA	O.TEST_OPERATE*	O.DLV_PROTECT	O.DLV_AUDIT	O.DLV_RESP*	O.USE_DIAG*	O.APPLI_DEV
T.DIS_DEL1	X						
T.DIS_DEL2		X					
T.MOD_DEL1	X						
T.MOD_DEL2		X					
A.DLV_PROTECT*			X				
A.DLV_AUDIT*				X			
A.DLV_RESP*					X		
A.USE_TEST*		X					
A.USE_PROD*		X					
A.USE_DIAG*						X	
A.APPLI_CONT							X

Table 10 : Mapping of security objectives for the environment to threats, assumptions and OSP relative on delivery from phase 1 to phases 4 to 6

T.DIS_DEL1

Unauthorized disclosure of ES data during delivery is countered by O.DLV_DATA, which specifies a trusted delivery maintaining the confidentiality.

T.DIS_DEL2

Unauthorized disclosure of and ES data after delivery is countered by O.TEST_OPERATE* which specifies maintenance of the confidentiality.

T.MOD_DEL1

Unauthorized modification of ES data during delivery is countered by O.DLV_DATA, which specifies a trusted delivery maintaining the integrity.

T.MOD_DEL2

Unauthorized modification of ES data after delivery is countered by O.TEST_OPERATE* which specifies maintenance of the integrity and its test.

A.DLV_PROTECT*

Procedures shall ensure protection of TOE material/information under delivery and storage.

A.DLV_AUDIT*

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage.

A.DLV_RESP*

Procedures shall ensure that people dealing with the procedure for delivery have got the required skill.

A.USE_TEST*

It is assumed that appropriate functionality testing of the TOE is used in phases 4, 5 and 6.

A.USE_PROD*

It is assumed that security procedures are used during all manufacturing and test operations through phases 4, 5, 6 to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

A.USE_DIAG*

It is assumed that secure communication protocols and procedures are used between Smartcard and terminal.

A.APPLI_CONT

Whenever a Loaded-Application is to be loaded on the platform, it is assumed that its development and production follow the Administrator Guidance.

IDMotion V2 Virtual Machine Security Target

Threats/Assumption/ OSP	O.LOAD	O.SECURITY	O.REMOVE	O.CIPHER	O.APPLI_DEV	O.DECIPHER
T.LOAD_MAN	X					
T.LOAD_APP	X					
T.LOAD_MOD		X			X	
T.APP_DISC		X			X	
T.APP_REMOVE			X			
T.DIS_DATA					X	
T.ERR_REMOVE			X			
T.CLON					X	
A.APPLI_CONT					X	
OSP.CIPHER				X		
OSP.CONF-ALU						

Table 11: Mapping of security objectives for the environment to threats, assumptions and OSPs on phases A1 and A2 (development and delivery for phase A1 to phases 6 and 7)

The TOE shall use state of the art technology to achieve the following IT security objectives and enforce the OSP.

T.LOAD_MAN

O.LOAD imposes that application be loaded only with the permission of the administrator, which counters the threat.

T.LOAD_APP

O.LOAD controls the origin of the Loaded Application before loading, thus if necessary control is made by the administrator, it counters T.LOAD_APP.

T.LOAD_MOD

Alteration of Loaded Application during loading is prevented by O.SECURITY, which guarantees its integrity. Modification of Code and data of a Loaded-Application during its transfer and loading is countered by O.APPLI_DEV, which ensures the mechanisms to verify their integrity

T.APP_DISC

Divulgence of Loaded Application during loading is prevented by O.SECURITY, which guarantees its confidentiality. Gaining access to confidential code and data of a Loaded-Application during its transfer and loading is countered by O.APPLI_DEV, which ensures the confidentiality.

T.APP_REMOVE

Removal of application without the consent of the administrator is countered by O.REMOVE, which imposes the authorization of the administrator.

T.DIS_DATA

Disclosure of sensitive User and TSF data is countered by is countered by O.APPLI_DEV, which requires that such data is stored encrypted.

T.ERR_REMOVE

Removal of application leaving confidential data is countered by O.REMOVE which imposes that the space left does not hold any information linked to remove application.

T.CLON

IDMotion V2 Virtual Machine Security Target

Cloning may require the disclosure of sensitive User Data. Disclosure of such data is countered by i O.APPLI_DEV, which requires that such data is stored encrypted.

A.APPLI_CONT

Whenever a Loaded-Application is to be loaded on the platform, it is assumed that its development and production follow the Administrator Guidance.

OSP.CIPHER

Cryptographic functions are required to actually protect the exchanged information (O.CIPHER). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them.

OSP.CONF-ALU

KTU used to protect to encrypt sensitive sections of the ALU is decrypted by MULTOS (card) (O.DECIPHER).

5.3.3 Assumptions and Security Objectives for the Environment

This section demonstrates that the combination of the security objectives upholds or satisfies the identified assumptions for the operational environment.

Each of the assumptions for the environment is addressed by objectives.

Table 8 demonstrates which objectives contribute to the satisfaction of each assumption.

For clarity, the table does not identify indirect dependencies.

Phases	Assumptions/O bjectives	Delivery process for phases 4 to 7			Phases 4 to 6 O.TEST_OPERA TE*	Phase 7 O.USE_DIAG*	Phase A1 O.APPLI_DEV
		O.DLV_PROTECT*	O.DLV_AUDIT*	O.DLV_RESP*			
4 to 7	A.DLV_PROTECT*	X					
4 to 7	A.DLV_AUDIT*		X				
4 to 7	A.DLV_RESP*			X			
4 to 7	A.USE_TEST*				X		
4 to 7	A.USE_PROD*				X		
7	A.USE_DIAG*					X	
A1	A.APPLI_CONT						X

Table 12: demonstrates mapping of security objectives for the operational environment to assumptions

A.DLV_PROTECT*

Procedures shall ensure protection of TOE material/information under delivery and storage. This assumption is upheld by **O.DLV_PROTECT*** in the delivery process for phases 4 to 7.

A.DLV_AUDIT*

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage. This assumption is upheld by **O.DLV_AUDIT*** in the delivery process for phases 4 to 7.

A.DLV_RESP*

Procedures shall ensure that people dealing with the procedure for delivery have got the required skill. This assumption is upheld by **O.DLV_RESP*** in the delivery process for phases 4 to 7.

A.USE_TEST*

It is assumed that appropriate functionality testing of the TOE is used in phases 4, 5 and 6. This assumption is upheld by **O.TEST_OPERATE***.

A.USE_PROD*

It is assumed that security procedures are used during all manufacturing and test operations through phases 4, 5, 6 to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use). This assumption is upheld by **O.TEST_OPERATE***.

A.USE_DIAG*

It is assumed that secure communication protocols and procedures are used between Smartcard and terminal. This assumption is upheld by **O.USE_DIAG*** in phase 7.

A.APPLI_CONT

Whenever a Loaded-Application is to be loaded on the platform, it is assumed that its development and production follow the Administrator Guidance. This assumption is upheld by **O.APPLI_DEV** in phase A1.

5.3.4 Additional Assumptions and Security Objectives for the Environment

This section demonstrates that the combination of the security objectives upholds the identified assumptions for the operational environment.

Assumptions/Objectives	O.TAMPER_ES	O.SIDE	O.CLON	O.OPERATE*	O.DIS_MECHANISM2	O.DIS_MEMORY	O.MOD_MEMORY	O.LOAD	O.SECURITY	O.REMOVE	O.CIPHER	O.DECIPHER
A.TAMPER_ES	X											
A.SIDE		X										
A.CLON			X									
A.OPERATE				X								
A.DIS_MECHANISM2					X							
A.DIS_MEMORY						X						
A.MOD_MEMORY							X					
A.LOAD								X				
A.SECURITY									X			
A.REMOVE										X		
A.CIPHER											X	
A.DECIPHER												

Table 13: Mapping of additional security objectives for the operational environment to assumptions

A.TAMPER_ES

This assumption is upheld by **O.TAMPER_ES**.

A.SIDE

This assumption is upheld by **O.SIDE**.

A.CLON

This assumption is upheld by **O.CLON**.

A.OPERATE

This assumption is upheld by **O.OPERATE**.

A.DIS_MECHANISM2

This assumption is upheld by **O.DIS_MECHANISM2**

A.DIS_MEMORY

This assumption is upheld by **O.DIS_MEMORY**

A.MOD_MEMORY

This assumption is upheld by **O.MOD_MEMORY**.

A.LOAD

This assumption is upheld by **O.LOAD**.

A.SECURITY

This assumption is upheld by **O.SECURITY**.

A.REMOVE

This assumption is upheld by **O.REMOVE**.

A.CIPHER

This assumption is upheld by **O.CIPHER**.

A.DECIPHER

This assumption is upheld by **O.DECIPHER**.

6. EXTENDED COMPONENTS DEFINITION

None.

7. SECURITY REQUIREMENTS

This chapter describes the security functional requirements for the TOE and the security assurance requirements for the TOE.

7.1 S SUPPORTING SECURITY INFRASTRUCTURE

The following roles and responsibilities are assumed within the MULTOS security infrastructure

- a) **MULTOS Security Manager (MSM):** defines and polices the MULTOS security infrastructure and provides criteria and services necessary for MULTOS participants to operate within the infrastructure. It acts as Certification Authority for the security infrastructure. It is assumed only one MSM exists. The MSM must be trusted by all participants in the infrastructure
- b) **MULTOS Implementor:** the organization that implements a MULTOS version. The MULTOS Implementor is licensed by MAOSCO and provides its MULTOS version to the IC Manufacturer. The MULTOS Implementor requests the MSM to provide MSM Controls Data, although this may be delivered to the MCD Manufacturer or MCD Issuer.
- c) **Integrated Circuit (IC) Manufacturer:** manufacturer of silicon from which chips and smartcards are made. It is assumed the IC Manufacturer is trusted to perform its tasks correctly: This includes:
 - To perform limited FLASH memory injection,
 - The injection of diversified transport key (used by Initialization Mode). Security keys and data are provided by the MSM.
- d) **Gemalto: is included in the new scheme in order to perform** the final manufacturing step by injecting the iKMA keys and data into the FLASH memory

The initialized ICs are provided to MCD Manufacturers:

- e) **MULTOS Carrier Device (MCD) Manufacturer:** responsible for embedding the IC in its plastic carrier and for background printing on the card. The result is an initialized MCD. This operation is assumed not to be security sensitive. The MCD Manufacturer may also receive MSM Controls Data from the MSM and enable the MCDs. Initialized and enabled MCDs are provided to MCD Issuers.
- c) **MCD Issuer:** responsible for issuing to users the MCD itself. The MCD Issuer may also enable initialized MCDs, by loading MSM Controls Data received from the MSM onto the MCDs. MCD Issuers retain the ultimate authority over what applications are loaded on their MCDs. MCD Issuers register applications with the MSM, provide information related to the applications and receive application load and delete certificates from the MSM.
- d) **Application Writer:** licensed by MAOSCO to produce applications for MULTOS. Supplies applications under contract to Application Issuers.
- e) **Application Issuer:** an organization that wishes to offer an application to MCD Users. The Application Issuer agrees with an MCD Issuer that the application can be loaded onto MCDs belonging to the MCD Issuer.
- f) **Application Provider:** the organization that takes responsibility for an application, by certifying it with the organization's public key and encrypting it where necessary. The Application Provider is a role that can be performed by an Application Writer, Application Issuer or MCD Issuer, rather than necessarily, being an organization in its own right.
- g) **Application Loader:** responsible for performing the technical operation of loading applications onto MCDs. The Application Loader enters into an agreement with one or more Application Issuers and MCD Issuers for loading applications supplied by one or more Application Providers.
- h) **MCD User:** final user of the MCD.

The MSM authorizes potential MULTOS platforms (known as MA-cards). To receive MSM authorization, a platform must comply with criteria covering attributes of the platform itself and the procedures associated with its manufacture.

7.2 SECURITY FUNCTIONAL REQUIREMENTS (SFRs)

This section defines the functional requirements for the TOE using only functional requirement components drawn from the CC part 2. The assignment and selection operations are written in **bold style** for a better readability.

7.2.1 Security Audit Automatic Response (FAU_ARP)

7.2.1.1 FAU_ARP.1 Security alarms

FAU_ARP.1.1 Mute Iteration. The TSF shall take action to cause **the MCD to mute** upon detection of a potential security violation.

7.2.2 Security audit analysis (FAU_SAA)

7.2.2.1 Potential violation analysis

FAU_SAA.1.1 Mute Iteration. The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 Mute Iteration. The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of:

- **An application attempts to execute MEL code outside of its code space or the code space of the codelet that it calls**
- **An application attempts to access data outside of its data space or the Public data segment**

known to indicate a potential security violation.

b) none.

7.2.3 Access control policy FDP_ACC

7.2.3.1 FDP_ACC.2 Complete access control

FDP_ACC.2.1 Load Application SFP Iteration. The TSF shall enforce the **Load Application SFP** on **MULTOS ES and Application Load Certificate**, and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.1 Delete Application SFP Iteration. The TSF shall enforce the **Delete Application SFP** on **MULTOS ES and Application Delete Certificate**, and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2. 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.

7.2.4 Access control functions FDP_ACF

7.2.4.1 FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 Load Application SFP Iteration. The TSF shall enforce the **Load Application SFP** to objects based on **Unique Application Identifier present in the ALC, Unique Application Identifier of loaded-applications, History List.**

FDP_ACF.1.2 Load Application SFP Iteration. The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed. **See the table below.**

IDMotion V2 Virtual Machine Security Target

FDP_ACF.1.3 Load Application SFP Iteration. The TSF shall explicitly authorize access of subjects to objects based on the following additional rules. **See the table below.**

FDP_ACF.1.4 Load Application SFP Iteration. The TSF shall explicitly deny access of subjects to objects based on the table below.

Security attributes	Governing access rules	Authorizing access rules	Denying access rules
Unique Application Identifier present in the ALC and Unique Application Identifier of loaded-applications	Verify there is other application currently loaded on this MCD with the same Application Identifier.	Establish that there is no other application currently loaded on this MCD with the same Application Identifier. Application load process continues.	Establish that there is another application currently loaded on this MCD with the same Application Identifier. Application load process is aborted.
History list	Determine if the application is being re-load a second time on to this MCD, and whether that is permitted	If the application is being re-load a second time on to this MCD, and that is permitted. Application load process continues.	If the application is being re-load a second time on to this MCD, and that is not permitted. Application load process is aborted.

FDP_ACF.1.1 Delete Application SFP Iteration. The TSF shall enforce the **Delete application SFP** to objects based on **Unique Application Identifier present in the ADC, Unique Application Identifier of loaded-applications.**

FDP_ACF.1.2 Delete Application SFP Iteration. The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed. **See the table below.**

FDP_ACF.1.3 Delete Application SFP Iteration. The TSF shall explicitly authorize access of subjects to objects based on the following additional rules. **See the table below.**

FDP_ACF.1.4 Delete Application SFP Iteration. The TSF shall explicitly deny access of subjects to objects based on the table below.

Security attributes	Governing access rules	Authorizing access rules	Denying access rules
Unique Application Identifier present in the ADC and Unique Application Identifier of loaded-applications	Verify an application with the Application Identifier specified in the ADC is loaded on this MCD	Establish that an application with the Application Identifier specified in the ADC is loaded on this MCD. Application deletion process continues.	Establish that no application with the Application Identifier specified in the ADC is loaded on this MCD. Application deletion process is aborted.

7.2.5 Rollback (FDP_ROL)

7.2.5.1 FDP_ROL.1 Basic rollback

FDP_ROL.1.1. The TSF shall enforce **Load Application SFP** to permit the rollback of the **load of an application on the application's code and data.**

FDP_ROL.1.2. The TSF shall permit operations to be rolled back within **a failure occurs during loading of an application.**

7.2.6 Fail secure (FPT_FLS)

7.2.6.1 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:

- a) **An application attempts to execute MEL code outside of its code space or the code space of the codelet that it calls**

b) An application attempts to access data outside of its data space or the Public data segment

7.2.7 Trusted recovery (FPT_RCV)

7.2.7.1 FPT_RCV.4 Function recovery

FPT_RCV.4.1. The TSF shall ensure that the **following list of functions and failure scenarios** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

Security functions	Failure scenarios
Application Load Control SF	Reset/power down during command processing
Application Delete Control SF	Reset/power down during command processing
Reset Protection SF	Reset/power down during command processing or application execution

7.2.8 Resource allocation (FRU_RSA)

7.2.8.1 FRU_RSA.1 Maximum quotas

FRU_RSA.1.1. The TSF shall enforce maximum quotas of the following resources: **Flash Memory** that **applications, functions, codelets and primitives** can use **simultaneously**.

7.3 SECURITY ASSURANCE REQUIREMENTS (SARs)

The security assurance requirement level is EAL7.

7.4 SECURITY FUNCTIONAL REQUIREMENTS DEPENDENCIES

This section demonstrates that all dependencies between components of security functional requirements included in this PP are satisfied.

Table 10 lists all functional components including security requirements in the IT environment. For each component, the dependencies specified in Common Criteria are listed, and an indication if these dependencies are satisfied for this TOE.

Security functional requirements	Dependencies in CC part 2	Dependencies in TOE
FAU_SAA.1: Potential Violation Analysis	FAU_GEN.1	No
FDP_ACC.2: Complete Access Control	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1: security attributes based Access Control	FDP_ACC.1, FMT_MSA.3	FDP_ACC.2
FPT_FLS.1: failure with preservation of secure state	None	None
FPT_RCV.4: Function recovery	None	None
FAU_ARP.1: Security Alarms	FAU_SAA.1	FAU_SAA.1
FDP_ROL.1: Basic Rollback	FDP_ACC.1	FAU_ACC.2
FRU_RSA.1: Maximum quotas	none	none

Table 14: Functional dependencies in Multi-Application environment

Incompatible dependencies are explained as follows:

- Dependencies on FDP_ACC.1 are replaced by FDP_ACC.2 (a higher hierarchical component).
- The dependency of FAU_SAA.1 with FAU_GEN.1 is not applicable to the TOE; the FAU_GEN.1 component forces many security relevant events to be recorded (due to dependencies with other functional security components) and this is not achievable in a smartcard since many of these events result in card being in an insecure state where recording of the event itself could cause a security breach. It is then assumed that the function FAU_SAA.1 may still be used and the specific audited events will have to be defined in the ST independently with FAU_GEN.1.
- The dependencies of FDP_ACF.1 on FMT_MSA.3 are not applicable because the security attributes used in Load Application SFP and Delete Application SFP (i.e. Application ID in ALC and the history list) are not managed by the TOE but by MSM (Multos Security Manager).

7.5 SECURITY REQUIREMENTS RATIONALE

7.5.1 Security Functional Requirements Rationale

This section demonstrates that the combination of the security requirements (TOE and environment) is suitable to satisfy the identified security objectives and that it can be traced back to the security objectives.

7.5.1.1 SFRs Tracing Rationale

In this sub-section Table 10 traces and demonstrates which security functional requirement contributes to the satisfaction of each TOE security objective. For clarity, the table does not identify indirect dependencies.

IDMotion V2 Virtual Machine Security Target

Security Functional Requirements	O.FLAW	O.ROLLBACK	O.RESOURCE	O.EFFECT_LOAD	O.EFFECT_REMOVE	O.SEGREGATE
EAL7 requirements	X					
FAU_ARP.1			X			
FAU_SAA.1			X			
FDP_ACC.2						X
FDP_ACF.1				X	X	X
FDP.ROL.1		X				
FPT_FLS.1		X		X	X	X
FPT.RCV.4		X		X	X	
FRU_RSA.1			X			

Table 15: Mapping of security functional requirements and objectives

7.5.1.2 SFRs Justifications Rationale

This sub-section describes and justifies how the security objectives for the TOE are met by the security functional requirements.

O.FLAW*

The objective is met by formal design and high-quality testing as specified by EAL7 conformity requirements.

O.ROLLBACK

This objective is assured by the following:

The TOE is in a valid state before a loading of an application thanks to FPT_FLS.1 and FPT_RCV.4. In the case of a failure in the loading, FDP_ROL.1 allows the rollback of the application's code and data automatically.

O.RESOURCE

Resource preservation objective is assured by the following:

Maximum quotas: FRU_RSA.1 supported by FAU_ARP.1 (Security Alarms) and FAU_SAA.1 (Potential violation analysis).

O.EFFECT_LOAD

Separation of the Loaded-Applications is assured by the following:

Security attributes: The Unique Application Identifier identifies each application stored in the memory of the MCD (FDP_ACF.1). Indeed, each application is stored in an Application Pool Block (which contains its code and data) which is identified by the Unique Application Identifier.

And the following SFR which assure correct operation:

Failure with preservation of secure state: FPT_FLS.1.

Function recovery: Security functions involved in application load have the capacity to either complete or to recover to a consistent and secure state (FPT_RCV.4).

O.EFFECT_REMOVE

Separation of the unloaded applications from the other Loaded-Applications is assured by the following:

Security attributes: The Unique Application Identifier permits the identification of each application stored in the memory of the MCD (FDP_ACF.1). Indeed, each application is stored in an Application Pool Block, (which contains its code and data) which is identified by the Unique Application Identifier.

And the following SFR which assures correct operation:

Failure with preservation of secure state: If an application attempts to modify code or data of remaining Loaded-Applications, a secure state is preserved (FPT_FLS.1).

Function recovery: Security functions involved in application deletion have the property to either complete or to recover to a consistent and secure state (FPT_RCV.4).

O.SEGREGATE

Segregation of Loaded-Applications is assured by the following:

User data protection and security management: The Unique Application Identifier, contained in the ALC (FDP_ACF.1), is used to identify the Application Pool Block in which the application's code and data are stored. It is directly coupled to the application load process but is also a basic requirement of segregation of Loaded-Applications (FDP_ACC.2).

The TSF fulfilling the SFR are protected by:

Failure with preservation of secure state: If a failure occurs (Loaded-Application trying to read from or write to another's Loaded-Application's code or data without authorization), a secure state is preserved (FPT_FLS.1).

7.5.2 SARs and the Security Requirements Rationale

The evaluation assurance level 7 is required to demonstrate that the TOE fulfills the most stringent Common Criteria requirements. In particular, the TOE design is formally described and its behavior is formally proved to correctly ensure its security objectives such as the segregation of Loaded-Applications (thanks to ADV requirements). Furthermore, the TOE is protected against sophisticated software and physical attacks (thanks to AVA_VAN.5 requirements).

7.5.2.1 ADV_SPM.1 Formal TOE security policy model

The formally modeled security policies consist of the segregation of the application executions, in particular:

1. If an application does not use the delegation mechanism, then its execution is contained in its memory space, i.e. this application cannot access to the data of the other applications (co-existing) on the TOE.
2. If an application uses the delegation mechanism, then its execution is contained in the memory space of the applications that have the delegation relation with it. In other words, this application cannot access to the data of another application that is not related to it by the delegation relation.

This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_SPM.1	SPM formal model and proof	Coq code
	General formalization approach for EAL7 requirements	Document
	Informal description of SPM	Document
	Fulfillment of ADV_SPM.1 requirements	Document
	Correspondence proof between the FSP and the SPM models	Coq code
	Informal description of SPM-FSP correspondence proof	Document

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

This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_FSP.6	FSP formal model	Coq code
	Informal description of FSP	Document
	Fulfillment of ADV_SPM.1 requirements	Document

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

This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_TDS.6	Formal model of (AM and MM) subsystems	Coq code
	Correspondence between subsystem and FSP models	Coq code

IDMotion V2 Virtual Machine Security Target

	Informal description of subsystem formal model	Document
	Informal description of FSP-subsystem correspondence proof	Document
	Correspondence proof between the subsystem formal model and the implementation	Document

7.5.2.4 ADV_IMP.2 Implementation of the TSF

This component is reused from EAL5+ evaluation because the TSF is included in EAL5's ADV_IMP.1 evaluation task.

7.5.2.5 ADV_INT.3 Minimally complex internals

This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_INT.3	Minimally complex internals	Document

7.5.2.6 ATE_DPT.4. Testing: implementation representatio

It depends on ADV_ARC.1, ADV_TDS.4, ADV_IMP.1 and ATE_FUN.1 that are all satisfied by this evaluation. This component is fulfilled by the following evident elements.

Requirement	Title	Type
ATE_DPT.4	Augmented Test Requirements and Formal Models	Document
	Collection of Evidences on Test coverage	Document

7.5.2.7 ATE_COV.3. Rigorous analysis of coverage

It depends on ADV_FSP.2 and ATE_FUN.1 that are all satisfied by this evaluation. This component is fulfilled by the following evident elements.

Requirement	Title	Type
ATE_COV.3	Augmented Test Requirements and Formal Models	Document
	Collection of Evidences on Test coverage	Document

7.5.2.8 ATE_FUN.2. Ordered Functional Testing

It depends on ATE_COV.1 that is satisfied by this evaluation. This component is fulfilled by the following evident elements.

Requirement	Title	Type
ATE_FUN.2	Augmented Test Requirements and Formal Models	Document

7.5.2.9 ALC_CMC.5 Advanced support

It depends on ALC_CMS.1, ALC_DVS.2 and ALC_LCD.1 that are all satisfied by this evaluation. This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_CMC.5	Evidence elements for ADV_CMC.5, ALC_LCD.2 and ALC_TAT.3	Document

7.5.2.10 ALC_LCD.2 Measurable life-cycle model

This component has no dependency and is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_LCD.2	Evidence elements for ADV_CMC.5, ALC_LCD.2 and	Document

IDMotion V2 Virtual Machine Security Target

	ALC_TAT.3	
--	-----------	--

7.5.2.11 ALC TAT.3 Compliance with implementation standards – all parts

It depends on ADV_IMP.1 that is satisfied by this evaluation.
This component is fulfilled by the following evident elements.

Requirement	Title	Type
ADV_TAT.3	Evidence elements for ADV_CMC.5, ALC_LCD.2 and ALC_TAT.3	Document

7.5.2.12 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.

7.5.2.13 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 smart card-based products hosting sensitive applications. AVA_VAN.5 is satisfied by reusing the corresponding EAL5 evaluation task.

8. TARGET OF EVALUATION SUMMARY SPECIFICATION

The TOE summary specification describes how the TOE meets each SFR.

8.1 SECURITY FUNCTIONALITY

This following defines the TOE security functions. The *italic paragraph parts* correspond to *actions provided by the security functions* whereas the normal paragraph parts correspond to the context in which the security functions take place.

Note that cryptographic primitives are out of scope.

Table 16 shows how these security functions satisfy the TOE security functional requirements.

8.1.1 Application Load Control SF (SF.Load)

SF.Load checks if an application, which has been previously loaded and then, deleted, is authorized by the MSM to be reloaded. The ALC contains a value that indicates if reloading the application onto the same MCD is authorized. The value can be zero or a random number generated by the MSM. A value of zero means that the MSM has authorized multiple loads of the application.

SF.Load ensures that the application is not already loaded on the MCD. When an attempt is made to load the application, the AID (unique Application Identifier) contained in the ALC is checked against the AID associated with each application already loaded on the MCD. If a match is found, this indicates the application has already been loaded onto the MCD and the load attempt will fail.

When loading the ALU components in the Application Pool Block in Flash memory, SF.Load checks if there is enough space available. If it is not the case, SF1 returns an error.

If load application fails, SF.Load ensures that the temporary loaded-application is erased on the next reset.

Permutational/probabilistic/cryptographic mechanisms used in this security function: None.

8.1.2 Application Delete Control SF (SF.Delete)

SF.Delete checks that the Application ID extracted from the Application Delete Certificate matches to a loaded application. Otherwise, no deletion is done.

Permutational/probabilistic/cryptographic mechanisms used in this security function: None.

8.1.3 Application Execution Management SF (SF.Firewall)

SF.Firewall ensures each application is restricted to accessing its own code and data. The only exceptions to the restriction on an application's code and data access are as follows:

- a) Accessing data in the Public Data Area.
- b) Application delegation.
- c) Accessing Codelets.
- d) Accessing data via MULTOS primitives.

SF.Firewall maintains separate storage and execution space for applications loaded onto an MCD. SF.Firewall manages a pool of loaded applications. SF.Firewall ensures each application, including its code and data areas, is kept separate from every other application loaded on the MCD. This ensures an application that is restricted to its own code and data space cannot gain access to the code or data of another loaded application. Each application is allocated to its own Application Pool Block within the Application Pool. Each Application Pool Block contains a unique identifier of the application loaded into the block. The intent of this mechanism is to allocate a portion of EEPROM memory to an

IDMotion V2 Virtual Machine Security Target

application where that portion does not overlap regions allocated to any other applications and to tag these regions of memory with the application ID of that application.

An application is able to read code for execution only from its own code space or from a pool of common routines controlled by SF.Firewall. SF.Firewall executes only applications written in the MULTOS Execution Language (MEL). MEL is an interpreted language. MEL applications are executed on an Application Abstract Machine, which enables memory accesses by applications to be checked at the time of interpretation. SF.Firewall ensures any attempt by an application to access code for execution is restricted to its own code space or to Codelets, which are controlled by SF.Firewall. This ensures an application is unable to compromise the integrity or confidentiality of the code of other applications loaded on the MCD.

SF.Firewall ensures no application is able to write to the code space of any application, including itself. SF.Firewall ensures any attempt by an application to write data is restricted to the application's own data space or to the Public data area. Any attempt by the application to write data outside these areas, including to its own or another applications code space, is blocked by SF.Firewall and the application is terminated.

SF.Firewall ensures no application is able to read from or write to the data space of another application except via a mechanism provided and controlled by SF.Firewall. SF.Firewall ensures any attempt by an application to read or write data is restricted to the application's data space or to the Public data area. The Public data area is available for reading and writing by all applications and provides the mechanism for applications to communicate information with each other. This ensures an application is unable to compromise the integrity or confidentiality of the data of other applications loaded on the MCD.

No application is able to cause the execution of another application except via a mechanism provided and controlled by SF.Firewall. SF.Firewall also provides a mechanism for an application to delegate execution to another application. On delegation, a full context switch occurs, so the only information from the delegating application which is available to the delegated application is whatever might be held in the Public data area. "Full context switch" means that SF.Firewall writes all information related to the execution of the delegating application to an area under its control then begins the execution of the delegated application. When the delegated application ends its execution, the execution context of the delegating application is restored and it is able to continue the execution from the point of delegation.

This ensures an application cannot make use of another application to compromise the integrity or confidentiality of other applications. Applications execute only within their own environment and cannot be made to execute in another application's environment.

SF.Firewall ensures no application is able to write to the code space of MULTOS and no application is able to read from or write to the data space of MULTOS except via a mechanism provided and controlled by SF.Firewall. SF.Firewall provides system primitives that can be invoked by applications, which return to the application specific system data values and allow specific system data values to be updated. Any other attempt by an application to access MULTOS code or data is blocked by SF.Firewall. This ensures no application is able to compromise the integrity of MULTOS or the confidentiality of its sensitive information.

8.1.4 Start-up Initialization SF (SF.Init)

If the MCD is reset or loses power while MULTOS is processing a command or executing an application, SF.Init will perform the usual validity checks and initialization when MULTOS is restarted:

- a) *SF.Init erases the Public data area (to protect any sensitive information placed there by an application executing at the time of reset/power loss).*
- b) *SF.Init erases from the Application Pool any application in the Application Pool that is in the "opened" state (since the application load process has been interrupted).*
- c) *SF.Init initializes the Active Application Block to the shell application if any is present, or otherwise to a null value to indicate that no application is currently selected.*
- d) *SF.Init rolls back any uncommitted writes in the Data Item Buffer.*

The Data Item Buffer or Data Item Stack holds a "stack" of data item copies. Each data item copy held in this stack contains a copy of a particular Static data item which MULTOS has, or is in the process of, updated as the result of executing an application MEL instruction or primitive.

IDMotion V2 Virtual Machine Security Target

This data item stack also contains information that allows SF.Init to determine, for each data item copy in the stack, whether the source data item has been successfully and completely updated.

The data item copy contains the following items. These items are located within the data item copy in the order given, with the first item at the lowest address:

- Flags and byte counts that allow navigation through the data item stack to find the most recent data item copy, to create a new data item copy, or to determine whether the most recent data item copy is a copy of an item which is in the process of being updated.
- A pointer to the start of the data item which the data item copy refers to.
- A copy of the data item.

When a data item copy is created SF.Init marks it as ACTIVE and when the source data item is successfully and completely updated SF.Init marks the data item copy as USED. If the card is reset and SF.Init found an ACTIVE block on the stack, SF.Init will copy it back to its original location and mark it as USED.

At the end of initialization, MULTOS is in the Ready state, waiting to process commands from the IFD. It therefore returns to a known secure state following a reset or power-down/power-up.

8.1.5 Mapping between Security Functions and Security Functional Requirements

Table 12 shows how functional security requirements are met by the security functions. Note that each security function contributes to the satisfaction of at least one TOE security functional requirement.

Functional requirements/ Security functions	SF.Load	SF.Delete	SF.Firewall	SF.Init
FAU_ARP.1			X	X
FAU_SAA.1			X	X
FDP_ACC.2	X	X		
FDP_ACF.1	X	X		
FDP_ROL.1	X			X
FPT_FLS.1			X	
FPT_RCV.4				X
FRU_RSA.1	X		X	

Table 16: Mapping between security functions and security functional requirements

8.1.6 TOE Security Functions Rationale

The following explains how the security functions work together so as to satisfy the TOE security functional requirements.

FAU_ARP.1 Mute Iteration

This requirement is implemented by the same security functions than FAU_SAA.1 because these actions are induced by the events defined in FAU_SAA.1.

FAU_SAA.1 Mute Iteration

The TSF detects a potential violation as soon as one of the events listed in FAU_SAA.1 appears. The security functions that satisfy the requirement are presented in the table below:

IDMotion V2 Virtual Machine Security Target

Security Function	Event
SF.Firewall	An application attempts to execute MEL code outside of its code space or the code space of the codelet that it calls.
	An application attempts to access data outside of its data space or the Public data segment.

FDP_ACC.2 Load Application SFP Iteration and Delete Application SFP Iteration

SF.Load enforces the Load Application SFP. It covers therefore FDP_ACC.2 Load Application SFP Iteration.

SF.Delete enforces the Delete Application SFP. It covers therefore FDP_ACC.2 Delete Application SFP Iteration.

FDP_ACF.1 Load Application SFP Iteration and Delete Application SFP Iteration

SF.Load covers FDP_ACF.1 Load Application SFP Iteration because it enforces the whole set of rules defined in this requirement.

SF.Delete covers FDP_ACF.1 Delete Application SFP Iteration because it enforces the whole set of rules defined in this requirement.

FDP_ROL.1

SF.Load and SF.Init permit the rollback of loading process by erasing the application's code and data if the application's load fails.

FPT_FLS.1

a) and b) are implemented by SF.Firewall.

FPT_RCV.4

In the event of a reset, a power down, a power loss or an application mute, MULTOS will perform its validity checks and initialization before entering its Ready State. In this way, the TSF is recovered to a consistent and secure state. SF.Init implements this part of the requirement.

FRU_RSA.1

SF.Load checks if there is sufficient amount of EEPROM before loading the ALU components.

Abbreviations and Acronyms

Term	Description
ADC	Application Delete Certificate.
ALC	Application Load Certificate.
ALU	Application Load Unit.
APB	Application Pool Block.
ATR	Answer To Reset.
CC	Common Criteria (for Information Technology Security Evaluation, Version 2.1).
CM	Configuration Management.
DES	Data Encryption Standard (algorithm).
EAL	Evaluation Assurance Level.
EEPROM	Electronically Erasable Programmable Read Only Memory.
ES	Embedded Software
IC	Integrated Circuit.
IFD	Interface Device (to smartcard).
IT	Information Technology.
KTU	Key Transform Unit.
MAOSCO	MAOSCO refers to the MULTOS Consortium. The MULTOS Consortium controls the MULTOS specification and is responsible for advancing the MULTOS OS in all smartcard related markets.
MCD	MULTOS Carrier Device.
MEL	MULTOS Executable Language (application language).
MSM	MULTOS Security Manager.
OSP	Organisational Security Policies.
PP	Protection Profile.
RAM	Random Access Memory.
ROM	Read Only Memory.
RSA	Rivest-Shamir-Aldeman (algorithm).
SAR	Security Assurance Requirement.
SFR	Security Functional Requirement.
SFP	Security Function Policy.
ST	Security Target.
TOE	Target Of Evaluation.
TSC	TSF Scope of Control.
TSF	TOE Security Functions.
TSFI	TSF Interface.

Term	Description
TSP	TOE Security Policy.

Vocabulary

Term	Description
Embedded software	Software embedded in a smartcard IC. Embedded software may be in any part of the non-volatile memory of the IC.
Integrated circuit (IC)	Electronic component(s) designed to perform processing and/or memory functions.
Phases	Refers to the seven phases of the smartcard product lifecycle, as outlined in the "Smartcard Integrated Circuit Protection Profile."
I/O peripherals	Material components of the TOE that manage its inputs/outputs.
Smartcard	A card according to ISO 7816 requirements, which has a non-volatile, memory and a processing unit embedded within it.
Smartcard embedded software	Composed of embedded software in charge of generic functions of the smartcard IC such as operating system, general routines and interpreters (smartcard basic software) and embedded software dedicated to the applications (smartcard application software).