SM10

Security Target V1.2

SAMSUNG SDS SOC Development Center

March 15, 2006

SAMSUNG SDS SAMSUNG

Doc. Number	KC067-SP-0101
Doc. Version	1.2
Author	SAMSUNG SDS

REVISION STATUS

Revision	Date	Description of Change
1.0	2005.12.05	Initial submission for evaluation
1.2	2006.03.15	Revised edition

Table of Contents

1 Introduction 5

1.1 Security Target Identification 5

1.2 Security Target Overview 5

1.3 Common Criteria Conformance 6

1.4 Writing Rules 6

1.5 Terms and Definitions 6

1.6 Security Target constitution 10

2 Target of Evaluation Description 11

2.1 TOE Summary 11

2.2 Product Type 11

2.3 Roles of TOE 12

2.4 Scope and Boundary of TOE 12

2.4.1 TOE Functions 13

2.4.2 Physical Scope of TOE 13

2.4.3 Logical Scope of TOE 15

2.5 TOE Life Cycle 15

2.6 Life Cycle 15

2.7 MULTOS Security Infrastructure 16

2.7.1 Cryptographic Keys 18

3 TOE Security Environment 20

3.1 Assets 20

3.2 Assumptions 21

3.3 Threats 21

3.4 Organizational security policies 23

4 Security Objectives 24

4.1 Security Objectives for the TOE 24

4.2 Security Objectives for the Environment 25

5 IT Security Requirements 26

5.1 TOE Security Functional Rquirements 26

5.1.1 Security Audit 27

5.1.2 Cryptographic Support 28

5.1.3 User Data Protection 29

5.1.4 Identification and Authentication 31

5.1.5 Security Management 33

5.1.6 Protection of the TSF 36

5.2 TOE Assurance Requirements 39

5.2.1 Configuration Management (CM) 40

5.2.2 Delivery and Operation 41

5.2.3 Development 42

5.2.4 Guidance Documents 46

5.2.5 Life Cycle Support 47

5.2.6 Test 49
5.2.7 Vulnerability Assessment 51
6 TOE Summary Specification 54
6.1 TOE Security Functions 54
6.2 Assurance Measures 54
7 Protection Profile Claims 55
7.1 Protection Profile Reference 55
7.2 Protection Profile Tailoring 55
7.3 Protection Profile Additions 55
8 Security Objectives Rationale 56
8.1 Security Objectives Rationale 56
8.1.1 TOE Security Objectives Rationale 57
8.1.2 Security Objectives Rationale for the Environment 58
8.2 Security Requirements Rationale 60
8.2.1 TOE Security Requirements Rationale 61
8.2.2 TOE Assurance Requirements Rationale 62
8.3 Dependencies Rationale 64
8.3.1 Dependencies of the TOE Security Functional Requirement 64
8.3.2 Dependencies of Security Functional Requirements for IT Environment 65
8.3.3 Dependencies of the TOE Assurance Requirement 65
8.4 TOE Summary Specification Rationale 66
8.4.1 TOE Security Functions Rationale 66
8.4.2 Strength of Security Function Level Rationale 70
8.5 Protection Profile Claims Rationale 70
8.6 Rationale about Mutual Supportiveness between Security Function Components 71
8.6.1 Bypass 71
8.6.2 Unauthorized Alteration 71
8.6.3 Disablement 71
8.6.4 Detection 71
Annex 73

1 Introduction

This document is the Security Target (shortly, ST) of SAMSUNG SDS MULTOS Card SM10 product.

This section identifies the ST and the Target of Evaluation (shortly, TOE) and it suppiles summary of ST and evaluation criteria the TOE conforms.

1.1 Security Target Identification

- Security Target Title: SM10 Security Target
- Security Target Version Number: 1.2
- Author: SAMSUNG SDS SOC Development Unit
- Evaluation criteria : Korea IT Security Evaluation and Certification Scheme (September 22, 2005)
- Evaluation assurance level: EAL4+
- Protection Profile compliance: Smartcard Open Platform Protection Profile for Government V1.0(December 28, 2004)
- TOE: SAMSUNG SDS MULTOS SM10 R2

1.2 Security Target Overview

This document is the ST of “SAMSUNG SDS MULTOS SM10 R2” which is an open platform Smartcard OS MULTOS which is implemented upon IC (Integrated Circuit) chip.

This product has Application Abstract Machine(shortly, AAM) which supports a MULTOS programming language MEL(MULTOS Executable Language) and MEL Application Programming Interface(shortly, API) as well in order to make it easy to implement on-card applications.

The TOE separates applications logically and makes them possess a unique encryption key and signing key so that it can enhance reliability and security.

The TOE is based on CC EAL4+ level SASMSUNG Electronics IC chip S3CC9RB and S3CC9P9, though TOE consists of software parts, not underlying hardware. The TOE is in compliance with the Smartcard Open Platform Protection Profile for Government V1.0(December 28, 2004).

The TOE is compose of MULTOS AAM, OS(Operating System) and APIs.

- MULTOS OS communicates with underlying hardware platform, executes memory management, and communicates with MULTOS AAM. Also, MULTOS OS loads and deletes application, and sends and receives APDU(Application Protocol Data Unit) commands and responses.
- MULTOS AAM provides primitives and instructions composed of standard library functions ensuring that all MULTOS implementations can be executed in the same manner.

The security objectives of the TOE are to protect the TOE itself, TOE data, and important data from unauthorized exposure or modification.

TOE provides the following main security features:

- Logical separation of data between applications
- Secure application load with certificate verification
- Secure application delete
- Valid smartcard authentication and life cycle management
- Key installation functions

1.3 Common Criteria Conformance

The Security Target conforms to the following requirements:

- Common Criteria Part 2(Security Functional requirement)
- Common Criteria Part 3(the Assurance Requirements) of EAL4+ level

Augmented assurance components from EAL 4 are as follows:

- ADV_IMP.2 “Implementation of the TSF”
- ALC_DVS.2 “Sufficiency of security measures”
- ATE_DPT.2 “Testing: low-level design”
- AVA_VLA.4 “Highly resistant”

And, the minimum strength level for the TOE security functions is "SOF-high."

1.4 Writing Rules

The Security Target uses abbreviations; the notation, format, and writing rules follows the Korea IT Security Evaluation and Certification Scheme (shortly, Common Criteria). The Common Criteria allows selection, assignment, refinement, and iteration operations which can be executed in the Security Functional requirement. Each operation is used in the Protection Porfile.

Iteration

The use of a component more than once with varying operations. The result of iteration operation is represented by interation number with round bracketed, that is, (Iteration number).

Selection

The specification of one or more items from a list in a component. The result of selection operation is represented by *underlined italics*.

Refinement

The addition of details to a component. It is represented by **bold**.

Assignment

The specification of an identified parameter in a component. It is represented with square bracket, that is, [Assignment_Value].

1.5 Terms and Definitions

The terms used in the Security Target follow those of the Common Criteria in case they are same.

Object

An entity within the TSC that contains or receives information and upon which subjects perform operations.

Attack potential

The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

SOF (Strength-of-Function)

A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms.

SOF-high

A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organised breach of the TOE security by attackers possessing a high attack potential.

Iteration

The use of a component more than once with varying operations.

ST (Security Target)

A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

PP (Protection Profile)

an implementation-independent set of security requirements for a category of the TOEs that meet specific consumer needs.

User

Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Selection

The specification of one or more items from a list in a component.

Smartcard Terminal

A device which has a keypad, display, security module and Smartcard read/write functions.

Identity

A representation (e.g. a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Element

An indivisible security requirement.

Role

A predefined set of rules establishing the allowed interactions between a user and the TOE.

Operation

Make a component counter to the specified threats in the Common Criteria or make it satisfy to the specific security policy(ex. Iteration, assignment, selection, refinement)

External IT Entity

Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.

Threat Agent

Unauthorized user or external IT entity who makes threat like illigal access, modification and deletion to the asset.

Authorized User

A user who may, in accordance with the TSP, perform an operation

Authentication Data

Information used to verify the claimed identity of a user.

Assets

Information or resources to be protected by the countermeasures of a TOE.

Refinement

The addition of details to a component.

Organizational Security Policies

One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

Dependency

A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.

Subject

An entity within the TSC that causes operations to be performed.

Augmentation

The addition of one or more assurance component(s) from CC Part 3 to an EAL or assurance package.

Abstract Machine

An hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine. Underlying abstract machine is OS if TOE is application, but it is firmware or hardware if TOE is OS.

Component

The smallest selectable set of elements that may be included in a PP, an ST, or a package.

Class

A grouping of families that share a common focus.

TOE (Target of Evaluation)

An IT product or system and its associated guidance documentation that is the subject of an evaluation.

EAL (Evaluation Assurance Level)

A package consisting of assurance components from CC Part 3 that represents a point on the CC predefined assurance scale.

Family

A grouping of components that share security objectives but may differ in emphasis or rigour.

Assignment

The specification of an identified parameter in a component.

ALU (Application Load Unit)

A unit which applications are loaded to MULTOS cards as. An application load unit consists of code and data.

EEPROM (Electrically Erasable Programmable Read-Only Memory)

EEPROM is a non-volatile storage chip used in computers and other devices to store small amounts of volatile (configuration) data. When larger amounts of more static data are to be stored (such as in USB flash drives) other memory types like flash memory are more economic. SEEPROM, meaning Serial EEPROM, is an EEPROM chip that uses a serial interface to the circuit board.

Integrated Circuit Chip

A semiconductor for Smartcard functions, and it has mask ROM, EEPROM, RAM and I/O port.

KTU (Key Transformation Unit)

A Key Transformation Unit (KTU) is required when loading Confidential Application Load Units. The purpose of the KTU is to protect the keys used in making the ALU confidential. The KTU will normally be created as part of the data preparation / ALU generation process. During application loading the KTU is used by the card to decrypt the confidential ALU.

MCD (MULTOS Carrier Device)

ICC that carries MULTOS operating system.

MEL (MULTOS Executable Language)

The instruction set of the Application Abstract Machine, as defined in the MULTOS Developers Reference Manual

MULTOS (Multi-Application Operating System)

An interoperable IC chip operating system providing multiple-application management

MULTOS OS

A name of operating system usually implemented on ICC to operate multiple application in a highly secure manner. It also implies the scheme of management and operation for the life cycle of MULTOS carrier device. MULTOS employs an end-to-end trust architecture that places the Issuer in control of their card base.

RAM (Random Access Memory)

A type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. There are two basic types of RAM: dynamic RAM (DRAM), static RAM (SRAM). The two types differ in the technology they use to hold data, dynamic RAM being the more common type. Dynamic RAM needs to be refreshed thousands of times per second. Static RAM does not need to be refreshed, which makes it faster; but it is also more expensive than dynamic RAM. Both types of RAM are volatile, meaning that they lose their contents when the power is turned off.

ROM (Read-Only Memory)

A computer memory on which data has been prerecorded. Once data has been written onto a ROM chip, it cannot be removed and can only be read. Unlike main memory (RAM), ROM retains its contents even when the computer is turned off. ROM is referred to as being nonvolatile, whereas RAM is volatile.

TSF (TOE Security Functions)

A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TSP (TOE Security Policy)

A set of rules that regulate how assets are managed, protected and distributed within a TOE.

TSF Data

Data created by and for the TOE, that might affect the operation of the TOE.

TSC (TSF Scope of Control)

The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

1.6 Security Target constitution

Part1 is the introduction.

This section provides introductory information required for the Protection Profile Identification.

Part 2 is the TOE description.

This section defines TOE and describes TOE environment.

Part 3 is the TOE Security Environment.

This section describes the security aspects of the environment, where the TOE is intended to be used, and addresses the description of the assets to be protected, the threats, the organizational security policies and the assumptions. It also describes means of the TOE to counter threats technically.

Part 4 is the Security Objectives.

This section describes security objectives for the TOE and environment to deal with identified threats, and supports assumption and Organizational Security Policies.

Part 5 is the IT Security Requirements.

This section describes security functional requirements and assurance requirements to satisfy security objectives. This part is from requirements of the Common Criteria part 2 and 3.

Part 6 is the TOE Summary Specification.

This section explains the TOE security functions implemented.

Part 7 is the Protection Profile Claims.

This section shows compliance to Smartcard Open Platform Protection Profile for Government V1.0.

Part 8 is the Security Objectives Rationale

This section demonstrates that security issues are countered adequately and the IT security requirements for security objectives are complet and adequate.

Annex is composed of the followings:

Reference clarifies the materials referenced in this Security Target.

Abbreviation provides terms and abbreviations frequently used.

2 Target of Evaluation Description

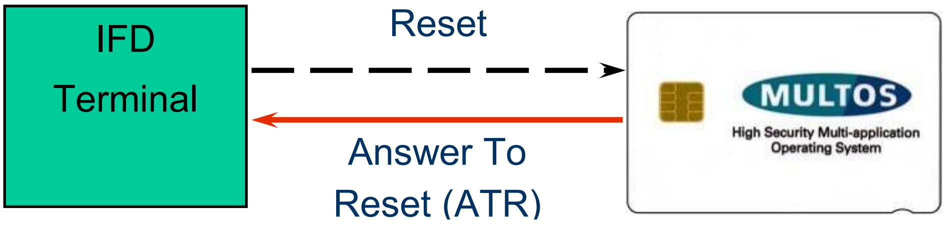
MULTOS is an operating system for integrated circuit cards(also known as smartcards). It is designed to allow multiple smartcard applications to be securely loaded and executed on a smartcard. It is an operating system for multiple applications smartcard providing high level of interoperability and security.

The TOE for Samsung SDS's SM10 is MULTOS that consists of MULTOS OS ROM Code and AMD (Additional MULTOS Data) embedded on both Samsung Electronics S3CC9RB and S3CC9P9 chip platforms. The TOE implements version 4.06 of the MULTOS specification and the following Change Requests approved by MULTOS consortium as well.

·CR 0027, CR0090, CR0103, CR0109, CR0113, CR0203

Because TOE has the same Application Programming Interface (API), applications can be executed in a same manner independent of a specific hardware.

In order for the TOE to load a smartcard application and to execute the application, the TOE uses a Card Acceptance Device(CAD) such as a dummy reader and/or an Interface Device(IFD) like POS(Point-of-Sale) terminal, Automatic Teller Machine(ATM) which supports ISO 7816 smartcard protocols.



An MCD (MULTOS Carrier Device) sends ATR(Answer-to-Reset) to an IFD when receiving Reset signal from the IFD. After that half-duplex communication starts between the IFD and MULTOS with command-response pairs, where a command is a message from the IFD to MCD and a response is a message from MCD to the IFD.

2.1 TOE Summary

The TOE of the security target is MULTOS that is an Open Platform being able to load multiple applications. The TOE has been embedded on the IC chip, enables communication with an IFD and provides the environment for application execution.

2.2 Product Type

As a multiple smartcard, SM10 is an MCD where the operating system implementing MULTOS specification has been embedded. The TOE contains APIs and the MEL interpreter used for executing MULTOS applications.

2.3 Roles of TOE

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 for 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).

MAOSCO is the organization which owns the rights to MULTOS. MAOSCO has an exclusive licence to develop and use the MULTOS specification. The principal security objectives of MULTOS are to protect the applications that are loaded onto MULTOS smartcards and to protect the control about MULTOS itself.

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.

The TOE is an operating platform to support multiple application programs, and achieves the following roles.

a) Execution of application program for MULTOS – performed under the lower hardware-independent environment.

b) Loading the multi-application programs. – Application programs must be able to exist together in a smartcard.

c) Assure for secure loading of application programs and seperation of each other – ensure no application loaded on the smartcard can interfere with the operation of any other loaded application or with MULTOS.

In other words, MULTOS provide the operating platform for application programs for smartcard and its development environment. Multiple application programs are loaded in a smartcard and not interfered with each other. And application programs are able to execute in other smartcard because MULTOS is hardware-independent platform.

2.4 Scope and Boundary of TOE

The Operating System or Platfrom Service can be used by the application program through Application Abstract Machine (AAM) which includes MEL API and MEL interpreter.

The TOE consists of components to perfrom the following functions.

- MULTOS Operating System

  MULTOS Operating System provides the communication with lower part, memory management and so on. MULTOS Operating System supports the functions for selecting applications, loading applicatons, deleting applications and communication by APDU command-response pairs.

- MULTOS AAM

MULTOS AAM provides the set of commands (called primitive) and library functions for executing same application on all the MULTOS implementation.

2.4.1 TOE Functions

MULTOS is the 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.

Applications to be loaded on MULTOS-based smartcards are written in the 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.

MULOS provides some features, as follows:

a) Appropriate authorization for applications

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.

b) Loading encrypted applications, decrypting such applicatons to make them available to the smartcard user for execution.

c) The capability to protect from interference about the operation of any other loaded application or MULTOS itself

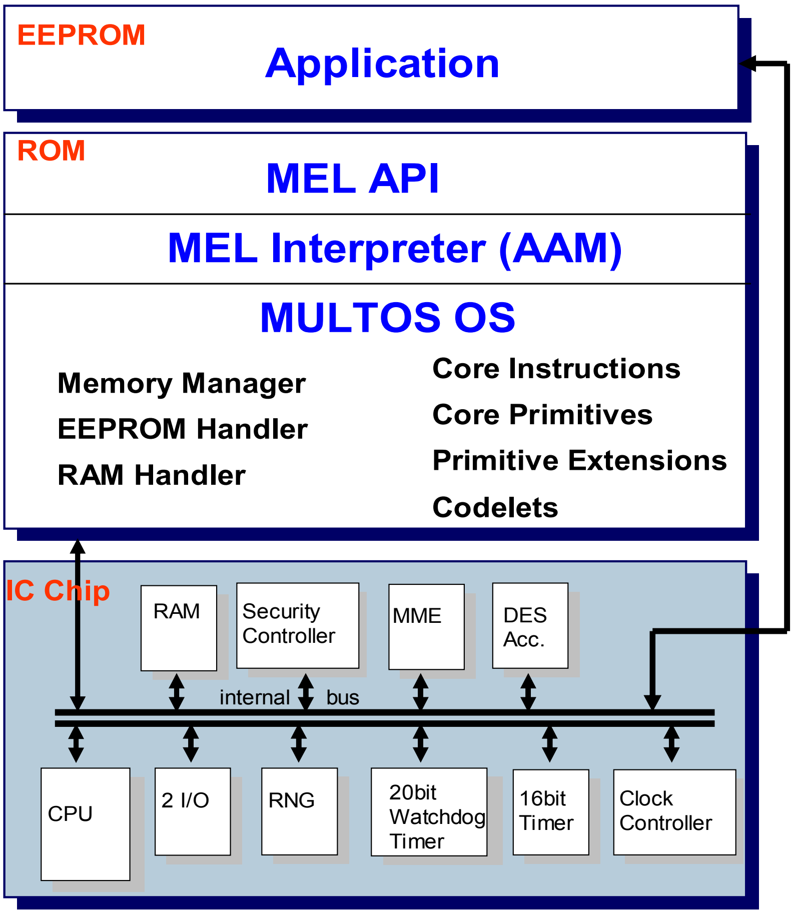
d) The capability to authenticate a card as a valid MULTOS equipped smartcard.

e) The capability to restrict the use of regulated features of the smartcard (e.g., strong cryptography) to authorized application.

f) Management of the number of failed attempts for the defined important function(installing keys, loading applications and deleting applications)

2.4.2 Physical Scope of TOE

MULTOS gets masked in ROM area of the underlying IC chip. MULTOS embedded in the IC chip performs mutual operations across the IFD-MULTOS interface.



TOE is implemented in the IC chip which is Samsung Electronics's S3CC9RB or S3CC9P9 certified CC EAL4+. The IC chip contains CPU that performs commands including executable code of MULTOS. Also it contains hardware for implementing the function of cryptography.

The IC chip is a physical processor in the smartcard. MULTOS provides independent platform about MULTOS application by executing source code and Abstract Machine in the IC chip

MULTOS requires firmware run-time libraries to support writing data to EEPROM. These libraries are supplied by Samsung Electronics. They provide low-level routines to support writing data to EEPROM, which is used on the target smartcard for the storage of applications.MULTOS requires firmware run-time libraries to support controlling registers of 512bit and 1024bit. These provide the function of cryptography for MULTOS.

The IC chip is composed of processor (CPU), random number generator (RNG), ROM, RAM, non-volatile EEPROM, I/O port, timer, etc from the physical point of view.

The application programs are loaded in the EEPROM by MULTOS. The smartcard that MULTOS is implemented is called MCD (MULTOS Carrier Device). Through each step of implementing MULTOS, an MCD-unique identifier (the MCD id) and MCD-unique symmetric transport key (tkv) that MSM Security Manager(MSM) knows are injected into non-volatile memory.

2.4.3 Logical Scope of TOE

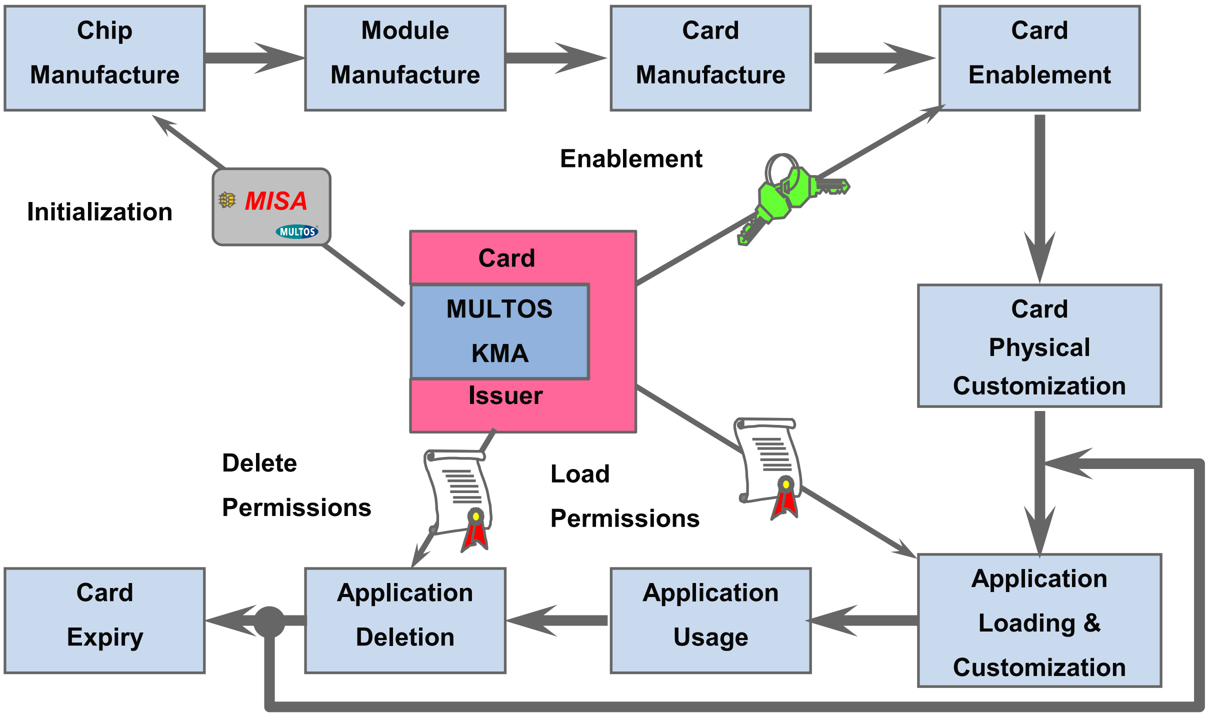
TOE is composed of the following modules logically.

- MULTOS Function Module – performs high level functions by communicating with IFD.
- MULTOS Memory Management – divides EEPROM memory into blocks and provides memory alteration function
- Secure Writing Module – provides controls to write all data on the EEPROM and atomicity
- Crypto Module – provides cryptographic functions used in MEL application program.
- AAM (Application Abstract Machine) – separation of MEL applications.

2.5 TOE Life Cycle

2.6 TOE Life Cycle

The following diagram shows general life cycle of the MULTOS card; role in each step is as follows.



- **MULTOS OS Developer**

  The MULTOS OS Developer is responsible for developing MULTOS OS and implementing code support the requirement that MAOSCO define in the specific IC Chip to provide from the IC Manufacturer.
- **Chip Manufacturer**

The Chip Manufacturer is responsible for embedding ROM code receiving from MULTOS OS Developer in the IC chip. MULTOS CA (KMA) provides the transport key and chip id of each chip to Chip Manufacture.

- **Module Manufacturer**

  The Module Manufacturer's role is to cut and make the smartcard chip of wafer form to the type of COB (Chip On Board).

- **Card Manufacturer**

  The Card Manufacturer is responsible for manufacturing MULTOS card as requirements of Card Issuer. The Card Manufacturer can be enabling MULTOS card and loading the application. In the step of Card Enablement, the Card Manufacturer activates the card so that it can load the application and load MSM Data and specific data that are provided from the Card Issuer.

- **Card Issuer**

  The Card Issuer is responsible for issuing to users the MCD itself. The Card Issuer is concerned about the whole process of card manufacture to card issue and takes charge of most of processings after issue (the expiry of card).

- **Application Writer**

  The Application Writer develops an application program according to specification and requirement of the Application Issuer. The Application Writer is licensed by MAOSCO to produce applications for MULTOS.

- **Application Issuer**

  An organization which wishes to provide an application to MCD Users, The Application Issuer agrees with a MCD Issuer that the application can be loaded onto MCDs belonging to the MCD Issuer.

- **Application Provider**

  The Application Provider takes responsibility to support for the application and manage necessary element (ex, application code, application data, key, and data) for loading the application.

- **Application Loader**

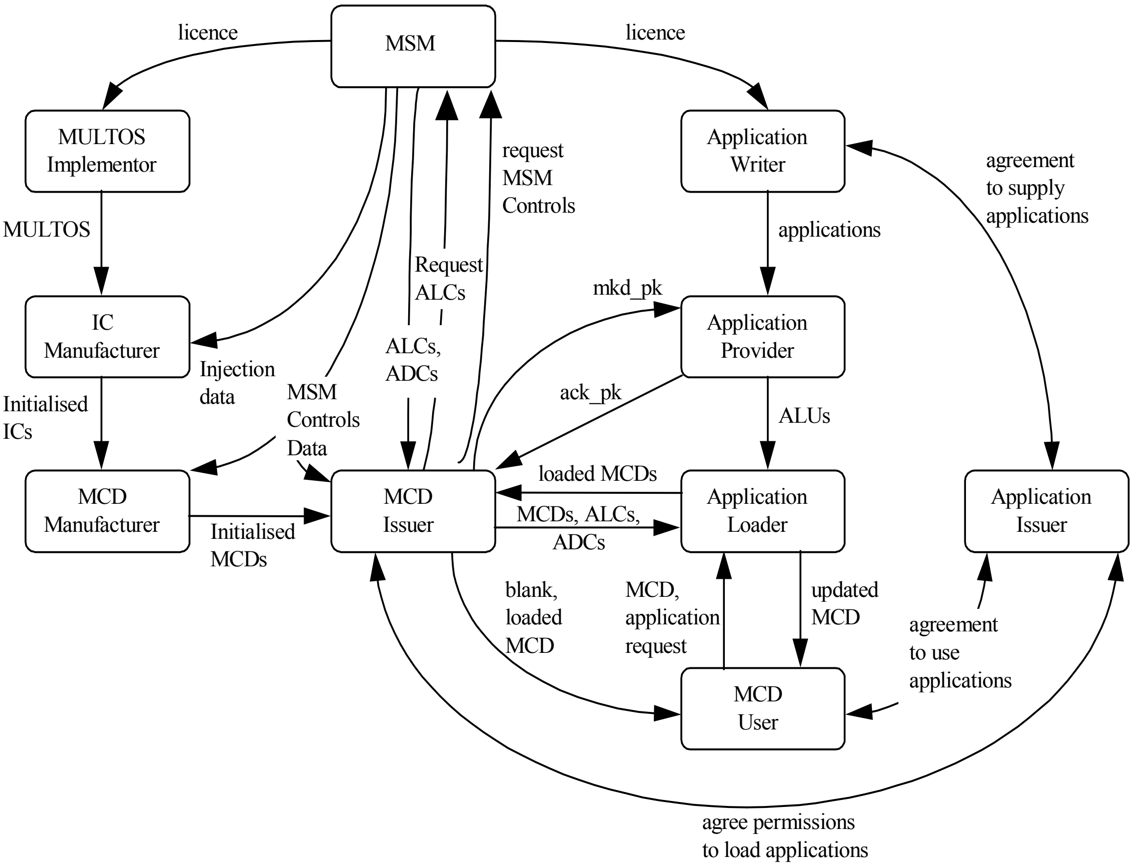
  The application loader performs the technical opertation of loading application and personalizaton data according to Application Provider's direction(security policy and procedure which made by Card Issuer)

- **MCD User**

  An MCD User is the User that use the card or the service to provide from Card Issuers and Application providers.

2.7 MULTOS 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.



MULTOS Infrastructure Context Diagram

- KMA

  The KMA is Key Management Authority managing to keys that is necessary to loading, deleting of application in the MULTOS.

- ALC(Application Load Certificate) or ADC(Application Delete Certificate)

  The ALC and ADC are Certificates that have the signature of MULTOS CA (Certification Authorization) for loading and deleting the application. By these certificates, MULTOS can prevent that applications are loaded or deleted to MULTOS card without Issuer's permission. When the application is loaded, The ALC, which is loaded to card with ALU, uses to allow loading the application.

- ALU(Application Load Unit)

  ALU contains the code and data of application. Generally the part of code includes a program or interger and the part of data includes parameter or personal information. ALU can include FCI, DF, KTU and application signature additionally. MULTOS loads this ALU in card to one data block. ALU is defined three kinds such as following contents.

  - Unprotected ALU
  - Protected ALU
  - Confidential ALU

- KTU(Key Transformation Unit)

The KTU is an element of an ALU using the confidential ALU. The KTU contains the DES keys and other information related to the manner in which a protected application has been encrypted. It is encrypted using the MCD-specific public key.

- MISA(MULTOS Injection Security Application)

  An MULTOS application that is distributed by the KMA as a secure means to inject the 64bytes of security data into each live MCD

2.7.1 Cryptographic Keys

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

[Table 1] MULTOS Security Infrastructure Keys

Key Name	Type	Role
kck	asymmetric	**Global Key Certification Key**
kck_sk		Held securely by MSM ; used by MSM to certify ADCs, ALCs(and indirectly through these, Application Provider public keys(ack_pk))
kck_pk		Held in ROM of every MCD ; used by MULTOS to verify ALCs, ADCs and Application Provider public keys.
ack	asymmetric	**Application Provider's asymmetric key** ; generated by Application Provider
ack_sk		Held by Application Provider ; used by Applicatoin Provider to sign application certificate.
ack_pk		Provided to MCD Issuer, who gets it certified by the MSM when ALCs and ADCs are requested.
tkck	asymmetric	**Transport Key Certification Key**
tkck_sk		Held securely by MSM ; used by MSM to certify MCD specific public transport keys(mkd_pk).
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).
tkv	symmetric	MCD-specific transport key ; generated by MSM ; stored in non-volatile memory of target MCD ; used by MSM to encrypt MCD-specific MSM Controls Data and also by MULTOS to decrypt the MSM Controls Data.

Key Name	Type	Role
mkd	asymmetric	MCD-specific asymmetric transport key
mkd_sk		Held in non-volatile memory of target MCD; used by MULTOS to decrypt KTU.
mkd_pk		Held by MSM; stored in non-volatile memory of target MCD; copy provided to Application Providers; used by Application Providers to encrypt KTU 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.
tkf	symmetric	Fixed part of MCD-specific transport key ; generated by MSM ; stored in non-volatile memory of MCD ; used by MSM, MCD Issuer and Application Loader to check authenticity of target MCD ; tkf is fixed for all MCDs.
misa_mk	symmetric	MISA Master Key; generated by MSM; used by MSM to generate misa_bk.
misa_bk	symmetric	MISA Base Key (each key value is unique to a given MISA). Used by MISA and MSM to determine tkv for a specific MCD.
hm	asymmetric	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.

3 TOE Security Environment

The security environment of the TOE consists of assumptions that describe the security aspects of the environment in which the TOE will be used, all threats that can be imposed for assets and environment of the TOE by threat agent and the organizational security policies that are rules, formality, traditional practice, guideline that the TOE must follow for security.

The smartcard is possessed and used freely by each card holder without physically access-controlled device, thus there can exist threats through logical interface as well as physical security threat that card itself is exposed to malicious environment.

3.1 Assets

The TOE operates on the IC chip and is a smartcard operating platform that manages the information and the resource. MULTOS is intended to provide a hardware-independent environment for the execution of multiple applications that provide a variety of functions and services.

Assets protected by the TOE and environment are composed of “Prinary assets“ and “Secondary assets.“

Primary assets that the TOE must protect are data managed in the smartcard. There are two types of data which are user data and the TSF data required for the operation of the TOE. And the documents created in the process of TOE production is additional assets that must be protected because they can affect to the integrity and the confidentiality of the TOE.

- Primary Assets:

  User Data protected by the TOE:

  - User Data: User Data that the TOE protect is the data that used by loaded application in the MCD or application program itself.
  - System Data: MULTOS file system and files, Initialization data, Manufacturer data, the TSF Data to protect these data (TOE Security Data such as Authentication data and Security attributes)
  - MULTOS Initialization Security Data:
    - kck_pk and hm that is stored in the MULTOS ROM.
    - Security Data (unique identifier), MCD specific Symmetric Transport Key (tkf and tkv), initialization date, security flag that is injected into non-volatile memory.
  - MSM Controls Data:
    - MCD specific Asymmetric Transport key (mkd)
  - Application Load Certificate(ALC), Application Load Units(ALU)

- Secondary Asset:

  The Secondary assets are information that is used to protect the integrity or the confidentiality of the TOE during the development step.

Also, because smartcard is a product that users possess and use, it can be the attacker's target to steal. Thus, IC chip itself is assets to be protected from physical threats.

The TOE is not an asset that must be protected directly, but the information that is produced and used during the process of the TOE production affects greatly in the integrity and confidentiality of the TOE itself. This information is called additional assets, and the security of additional assets shall be assured by the requirement of EAL4+

3.2 Assumptions

It is assumed that the following terms exist in the TOE operation environment accepting this Security Target.

A.ATK_LEV

Attackers possess a high level of expertise, resources and motivation and have high possiblility that discovers vulnerability that can be abused.

A.SEC_CNL

There shall be a secure channel between the TOE and the IFD.

A.APP_INS

The application program must follow approved procedure when installed into the TOE. If the application program is installed adequately, it shall not contain malicious code.

A.UNDER_HW

Underlying hardware upon which the TOE is operated must be physically secure.

A.TOE_HNDL

In the steps from manufacturing to using the TOE, there are roles of manufacturers, issuers and holders and training to each role shall be conducted in accordance with defined provisions. And the TOE is handled in a secure manner when repaired or replaced due to breakdown of the TOE or the smartcard.

A.TSF_DAT

TSF data that are exposed to be processed in the course of TOE operation are managed securely.

3.3 Threats

Because the smartcard is possessed and used by individuals without physically-controlled device, physical threats as well as logical threats exist.

T.LOGI_ATK

The threat agent may abuse logical interfaces to modify and disclose user data or TSF data.

T.ISS_ABU

The threat agent may abuse the TOE while the smartcard including the TOE is issued.

T.UNAU_IFD

The threat agent may modify and disclose user data or TSF data by abusing an unauthorized terminal.

T.UNAU_APP

The threat agent may modify and disclose user data or TSF data by unauthorized installation of application program containing malicious code.

T.SES_TEAR

User data or TSF data can be disclosed or damaged by incomplete termination of TSF service due to cut of power supply or an impact during card usage.

T.REP_AUTH

The threat agent may access to the TOE by attempting authentication repeatedly.

T.ABN_END

The threat agent may modify and disclose user data or TSF data in case of incomplete termination of a TSF service caused by attacks based on giving physical stress to smartcard and.

T.RES_COL

The threat agent may modify and disclose user data or intrude into execution area of the other application to cause malfunction of smartcard.

T.RES_REU

When the TOE reuses resources, the threat agent may access to unauthorized information in case the object information is not removed adequately.

T.LEAK_INF

The threat agent may abuse the information leaked from the TOE during the normal TOE operation.

T.TAMPER

The threat agent may acquire user data and TSF data via accessing directly to the IC or memory using chemical materials or elaborate equipments.

3.4 Organizational security policies

Organizational security policies described this section must be observed in the TOE following this Security Target.

P.DUTY_SEP

Role must be assigned to the responsible personnel of each step from manufacturing to using the smartcard and the TOE must be created and managed according to each role in a secure manner.

P.CRYPTO

The TOE must use cryptographic algorithms and modules that are approved by the National Intelligence Service.

P.OPN_PLAT

The TOE must be developed as an open platform which can load and use multiple applications in an interoperable manner.

4 Security Objectives

This Security Target defines and describes the following security objectives:

- Security Objectives for the TOE
- Security Objectives for the environment

4.1 Security Objectives for the TOE

The followings are security objectives that shall be directly handled by the TOE.

O.DAT_PROT

The TOE shall ensure that only an authorized user can access and modify user data and TSF data.

O.AUTH_ISS

The TOE must ensure that authorized issure can issue smartcards according to defined procedure.

O.AUTH_USR

The TOE must clarify users who can use logical interfaces and their access rights to assets.

O.APP_SEP

The TOE must support separation of each application's execution area in order to prevent collision when resources between applications are shared.

O.AUTH_RPR

The TOE must ensure that only an authorized user can conduct fix the failure.

O.ACC_AUTH

A user must conduct authenticaion procedure before accessing TOE user data and TSF data.

O.ROLLBACK

The TOE shall be rolled back to a well-defined valid state in case of TSF failure. And the TOE must detect the failure of the TSF and resume TSF service from the state before failure.

O.REM_RES

The TOE shall ensure that user data or TSF data stored in memories is protected against an unintended remaining in case of session closed in an work area used by TSF.

O.CNR_LEAK

The TOE must be designed to prevent normally disclosed information from being abused.

4.2 Security Objectives for the Environment

The followings are security objectives that are handled by information technology or non-technical/procedural means.

OE.ATK_LEV

The attacker shall have high level of domain knowledge, resources and motivation while finding out abusable vulnerabilities and abusing them with high probability.

OE.SEC_CNL

A secure communication cannel shall be provided between the TOE and corresponding smartcard accepting device.

OE.TSF_DAT

The TOE must operate the TSF data processed out of the TOE in a secure manner.

OE.TRAINING

Administration training must be conducted for each role of step of manufacturing, issuing, and using .

OE.C_TAMPER

The TOE must provide means to counter attacks attempting to obtain user data or TSF data via accessing directly to circuits or memory of the IC chip using chemical materials or elaborate equipments.

OE.UNDER_HW

The TOE must ensure secure operation on a tamper-resistant IC chip, and the underlying hardware of the TOE shall provide means to counter various tampering attacks.

OE.APP_INS

The application installation must follow approved procedure, and adequately loaded applications shall not contain malicious code.

OE.OPN_PLAT

The smartcard must support an open platform where multiple applications can be loaded to use.

5 IT Security Requirements

IT Security requirements describe the security functional requirements and the desired security behaviour expected of a Target of Evaluation (TOE) expressed in a securiyt target.

The assurance level of the Security Target is EAL4+ (ADV_IMP.2, ALC_DVS.2, ATE_DPT.2, AVA_VLA.4). The lowest security functional strength is “SOF-high(functional strength-high)”. The Security Target assumes that the possibility of the successful attack of a threat is high. Thus, the required security functional strength is defined as high.

5.1 TOE Security Functional Rquirements

The sucurity functional requirements defined at the Security Target selected and used the functional components related to the one of the common evaluation criteria 2 in order to satisfy the security objectives identfied at the previous section. The security functional requirements consist of the components selected at the common evaluation criteria 2.

[Table 2] TOE Security Functional requirements

Security Function class	Security Function components	
Security audit	FAU_ARP.1	Security alarms
	FAU_SAA.1	Potential violation analysis
Cryptographic support	FCS_CKM.1	Cryptographic key generation
	FCK_CKM.4	Cryptographic key destruction
	FCS_COP.1	Cryptographic operation
User data protection	FDP_ACC.2	Complete access control
	FDP_ACF.1	Security attribute based access control
	FDP_RIP.1	Subset residual information
Identification and authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of confidentials
	FIA_UAU.1	Authentication
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.6	Re-authenticating
	FIA_UID.1	Identification
Security management	FMT_MOF.1	Management of security functions
	FMT_MSA.1	Management of security attributes
	FMT_MSA.2	Secure security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_MTD.1	Management of TSF data
	FMT_MTD.2	Management of limits on TSF data
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_AMT.1	Abstract machine testing

	FPT_FLS.1	Failure with preservation of secure state
	FPT_PHP.3	Resistance to physical attack
	FPT_RCV.3	Automated recovery without undue loss
	FPT_RCV.4	Function recovery
	FPT_SEP.1	TSF domain separation
	FPT_TST.1	TSF testing

5.1.1 Security Audit

FAU_ARP.1 Security alarms

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [the below list of actions] upon detection of a potential security violation.

list

1) the MCD to abend and become mute

2) the MCD to enter shutdown mode

Dependencies: FAU_SAA.1 Potential violation analysis.

FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

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

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

a) Accumulation or combination of [the below auditable events] known to indicate a potential security violation;

1) Abend(Abnormal end)

- 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.
- An apparent corruption of the MSM Controls Data or security data held within the EEPROM of MULTOS.
- An unexpected hardware event occurred.
- MULTOS determines that it has executed an invalid sequence of instructions (possibly due to electromagnetic or mechanical interference).
- An apparent corruption of an application’s code space held within the Application Pool Block in the EEPROM of MULTOS.

2) Shutdown(End)

- An EEPROM write fails.
- A critical process is interrupted.
- There have been too many failed attempts to load MSM Controls Data.

b) [*None*]

Dependencies: FAU_GEN.1 Audit data generation

5.1.2 Cryptographic Support

FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA key pair generation] and specified cryptographic key sizes [1024] that meet the following: [PKC#1].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [erasure of a temporary copy key present in RAM] that meets the following: [none].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FCS_CKM.1 Cryptographic key generation]
FMT_MSA.2 Secure security attributes

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1/DES

FCS_COP.1.1 The TSF shall perform [recovering of protected code or data segments of an application and recovering of MSM Controls Data] in accordance with a specified cryptographic algorithm [DES decryption] and cryptographic key sizes [8byte(single key) or 16byte(double key)] that meet the following: [the below list of standards.

- FIPS PUB 46-3, Data Encryption Standard(ANSI X3.92)
- FIPS PUB 81, DES Modes of Operation

FCS_COP.1/SEED

FCS_COP.1.1 The TSF shall perform [encryption, decryption] in accordance with a specified cryptographic algorithm [SEED] and cryptographic key sizes [128 bits] that meet the following: [the below list of standards].

- TTAS.KO-12.0004: 128-bit Symmetric Block Cipher SEED
- TTAS.KO-12.0025: Modes of Operation for The Block Cipher SEED

FCS_COP.1/RSA

FCS_COP.1.1 The TSF shall perform [digital signature verification] in accordance with a specified cryptographic algorithm [RSA decryption] and cryptographic key sizes [768 bits for modulus and 3 for public exponent] that meet the following: [the below list of standards

- PKCS#1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002.
- ANSI X9.31, PKCS#2 and IEEE-P13-63

FCS_COP.1/SHA-1

FCS_COP.1.1 The TSF shall perform [secure hash] in accordance with a specified cryptographic algorithm [SHA-1] and cryptographic key sizes [which are not applicable here because there is no key needed for a hash] that meet the following: [the below list of standards].

- ANSI X9.30, NIST FIPS180-2: Secure Hash Standard, October 2, 1995

FCS_COP.1/Asymmetric Hash

FCS_COP.1.1 The TSF shall perform [application code integrity checking and MCD authentication] in accordance with a specified cryptographic algorithm [Asymmetric Hash] and cryptographic key sizes [578 bits for hash modulus and 3 for public exponent] that meet the following: [the below list of standards].

- MULTOS-4 Architecture Specification Security Specification(mao-des-110), March 7, 2000 , The Asymmetric Hash function is based on RSA

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security

5.1.3 User Data Protection

FDP_ACC.2 Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

FDP_ACC.2.1 The TSF shall enforce the [Application access control SFP] on [the below list of subjects and objects] 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 in the TSC and any object within the TSC are covered by an access control SFP.

- subjects list
  - S.Application
  - S.MCD
- objects list
  - O.CODE: application reads code for execution
  - O.DATA: application access data
  - O.CODELET: application to execute shared code routines
  - O.PUBLIC: public data is accessible any application
  - O.SYSTEMDATA: can be read and written by application, via primitives

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [Application access control SFP] to objects based on the following: [Application states, application identifier, and S.MCD identifier].

Application access control SFP

- This SFP controls the following operations: load, install, and delete of an S.Application
- Every S.MCD, S.Application shall be uniquely identified
- S.Application stetes are modeled by ‘notpresent’, ‘opened’, ‘passive’, ‘active’, ‘delegated’

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

- The MSM shall authorize all requests to load and delete an S.Application
- Authorizing an application load request, the MSM shall also authorize the ability to subsequently reload the application if it is deleted from a smartcard
- S.Application shall be authenticated before it is loaded onto a MCD, unless specific authorization is given by the MSM
- An application loaded onto a MCD shall be able to read code for execution only from its own code space or from a pool of common routines controlled by MULTOS
- An application loaded onto a MCD shall not be able to write to the code space of any application on the MCD, including its own
- An application loaded onto a MCD shall not be able to read from the data space of any other application loaded onto the MCD except via a mechanism provided by and controlled by MULTOS, and with the co-operation of the target application
- An application loaded onto a MCD shall not be able to write to the data space of any other application loaded onto the MCD except via a mechanism provided by and controlled by MULTOS

- An application loaded onto a MCD shall not be able to read or write MULTOS data, except via a mechanism provided and controlled by MULTOS
- An application loaded onto a MCD shall not be able to write to the code space of MULTOS
- It shall not be possible to read data from an application after that application has been deleted from a MCD

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [None]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [None].

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [deallocation of the resource from] the following objects: [application's code and data spaces: Transient objects, Cryptographic buffers, Transaction buffer, APDU buffer].

Dependencies: No dependencies.

5.1.4 Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [6] unsuccessful authentication attempts occur related to [SetMSMControls command, DeleteMELApplication command and CreateMELApplication command].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [return an error, permanently disables the function].

Dependencies: FIA_UAU.1 Authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [the below list of security attributes].

User	Security attribute
MULTOS Security Manager	Global Key Certification Key (kck)
	MCD-specific Asymmetric Transport Key (mkd)
	MCD-specific Transport Key variable (tkv)

Application Provider	Application Provider's Asymmetric Key (ack)
MCD Issuer	MCD Issuer Identifier

Dependencies: No dependencies.

FIA_SOS.1 Verification of confidentials

Hierarchical to: No other components.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that confidentials meet [RSA generation key metric].

Dependencies: No dependencies.

FIA_UAU.1 Authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The TSF shall allow [Processing of Check Data Command] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user in addition to FIA_UAU.1.1

Dependencies: FIA_UID.1 identification

FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [application's load and delete authentication mechanisms].

Dependencies: No dependencies.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [*all request to delete applications*

Dependencies: No dependencies.

FIA_UID.1 identification

Hierarchical to: No other components.

FIA_UID.1.1 The TSF shall allow [processing of Check Data command] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall allow [Check Data Command for MCD prior to loading it with MSM Controls Data] on behalf of the user and require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user in addition to FIA_UID.1.1.

Dependencies: No dependencies.

5.1.5 Security Management

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [determine the behaviour of the behaviour of] the functions [of the below list] to [the below roles].

Role	SF	Behavior	Detailed behavior
Application Loader	Application Load and Authentication SF	enable	Load the contents of an Application Load Unit (ALU) onto the MCD
MCD Issuers	Smartcard Authentication SF	enable	determine that a MCD is an authentic initialised MCD prior to loading it with MSM Controls Data

Dependencies: FMT_SMR.1 Security roles

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [*Application access control SFP*] to restrict the ability to [*perform the following operations*] the security attributes [of the below list] to [*MSM*].

Object	Security attributes	Operation	SFP	Role
D.Application	MCD Issuer Identifier MCD Batch Number MCD-Unique Identifier Asymmetric transport key set	Load	Application Access control	Application Loader MCD Issuers

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles

FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

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

Dependencies: ADV_SPM.1 Informal TOE security policy model
[FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

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

FMT_MSA.3.2 The TSF shall allow the [MCD Issuer] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [inquire] the [below MULTOS Security Data] to [MCD Issuer].

- a unique identifier, based on the MISA identifier and ICC serial number
- 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.
- initialization date, indicating when the security data was injected into the MCD
- a security flag indicating MSM Controls Data has not been loaded.

Dependencies: FMT_SMR.1 Security roles

FMT_MTD.2 Management of limits on TSF data

Hierarchical to: No other components.

FMT_MTD.2.1 The TSF shall restrict the specification of the limits for [D.Maxtries] to [MCD permanently disable below functions].

Failed attempts to execute of

- Application Load and Authentication retry count
- Application Transport Confidentiality retry count
- Application Deletion retry count
- Key Installation retry count

FMT_MTD.2.2 The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [MCD disable].

Dependencies: FMT_MTD.1 Management of

FMT_SMR.1 Security roles

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [**defined in the following list**].

- **MULTOS OS Develpoer**

  is responsible for developing the MULTOS OS and implement the requirements that is specified by the MAOSCO for the chip supplied by an chip manufacturer.

- **IC Manufacture (Chip Manufacture)**

  is responsible for receiving the ROM code from the MULTOS OS developer and load it on the IC chip. The MULTOS CA (KMA) supplies the transport key and MCD_ID for each MCD to a chip manufacturer..

- **Module Manufacture**

  is responsible for cutting the wafer type of a smartcard IC chip and assembling the COB(Card On Board).

- **MCD Manufacture (Card Manufacture)**

  is responsible for manufacturing a MULTOS card according to the requirement of the card issuer. The enablement of a MCD and the loading an application are frequently done by the MCD manufacturer. On the enablement stage, the MCD manufacturer enables the MCD to load an application and loads some personalization data and issuer specified data received from the card issuer on the MCD.

- **MCD Issuer(Card Issuer)**

  is responsible for issuing a MULTOS card to a card user. Additionallly, the MCD issuer participates in the total process of manufacturing and is charge of all the affairs of the post-issued card such as the expiration of card.

- **Application Developer(Application Writer)**

  developes an application according to the requirements of the card issuer and specifications. The application developer should be granted the license of the MULTOS application development by MAOSCO

- **Application Issuer**

  supplies an application to the MCD user and plaies a role in asking the MCD issuer to consent to the load of an applicaiton

- **Application Provider**

  manages the components that are need to load an application on a card such as the application code, application data, key and data for the MCD user.The application provider plaies a role in supporting the MCD user for the application.

- **Application Loader**

  executes the technical process that loads an application and a personalization data on a card according to the instruction of the application provider, the security policies and processes established by the card issuer

- **MCD User**

  are those who use the card to be supplied by the card issuer and the related application provider and the service provided by the card.

FMT_SMR.1.2 The TSF shall be able to associate users with roles defined at the FMT_SMR.1.1.

Dependencies: FIA_UID.1 Timing of identification

5.1.6 Protection of the TSF

FPT_AMT.1 Abstract machine testing

Hierarchical to: No other components.

FPT_AMT.1.1 The TSF shall run a suite of tests [during initial start-up] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Dependencies: No dependencies.

FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [the below list]..

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

c) An apparent corruption of the MSM Controls Data or security data held within the EEPROM of MULTOS

d) An unexpected hardware event occurred

e) MULTOS determines that it has executed an invalid sequence of instructions (possibly due to electromagnetic or mechanical interference)

f) An EEPROM write fails

g) A critical process is interrupted

h) There have been too many failed attempts to load MSM Controls Data.

Dependencies: ADV_SPM.1 Informal TOE security policy model

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

FPT_PHP.3.1 The TSF shall resist [following physical tampering scenarios] to the [following TSF devices] by responding automatically such that the TSP is not violated.

Physical tampering scenarios	TSP devices/elements
Abnormal use of reset signal	All TSF devices/elements
Abnormal use of power signal	All TSF devices/elements
Clock rate variations	The processor
Dynamic power analysis	Cryptographic operations

Dependencies: No dependencies.

FPT_RCV.3 Automated recovery without undue loss

Hierarchical to: FPT_RCV.2 Automated recovery

FPT_RCV.3.1 When automated recovery from [failures or service discontinuities] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2 For [Power failure and communication failure], the TSF shall ensure the return of the TOE to a secure state using automated procedures

FPT_RCV.3.3 The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [100%] for loss of TSF data or objects within the TSC.

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

Dependencies: FPT_TST.1 TSF testing
AGD_ADM.1 Administrator guidance
ADV_SPM.1 Informal TOE security policy model

FPT_RCV.4 Function recovery

Hierarchical to: No other components.

FPT_RCV.4.1 The TSF shall ensure that [following 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 Certificate Control	Reset/power down during command processing
Application Delete Certificate Control	Reset/power down during command processing Too many failed Delete Command
Unprotected/Protected Application Load Unit	Reset/power down during command processing Too many failed Create Command

Confidential Application Load Unit	Reset/power down during command processing Too many failed Create Command
MSM Controls Data Load Management	Reset/power down during command processing Too many failed Set MSM Controls Command
Application Execution Management	Application abend Reset/power down during command processing or application execution
Critical Data Overwrite	Reset/power down during command processing or application execution
Reset Protection	Reset/power down during command processing or application execution
Integrity Checks	Reset/power down during command processing or application execution
Start-up Validity Checks and Initialization	Reset/power down during command processing or application execution
All Security Functions	EEPROM write failure Power loss Integrity failure

Dependencies: ADV_SPM.1 Informal TOE security policy model

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Dependencies: No dependencies.

FPT_TST.1 TSF testing

Hierarchical to: No other components.

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

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of [TSF data].

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

Dependencies: FPT_AMT.1 Abstract machine testing

5.2 TOE Assurance Requirements

The assurance requirements of this Security Target are composed of assurance component in the Common Criteria Part 3 and assurance level is EAL4+. [Table 3] shows summarized assurance components.

Newly added assurance components in the Security Target are as follows.

- ADV_IMP.2 Implementation of the TSF
- ALC_DVS.2 Sufficiency of security measures
- ATE_DPT.2 Testing: low-level design
- AVA_VLA.4 Highly resistant

[Table 3] TOE the Assurance Requirements

Assurance Class	Assurance component	
Configuration management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
Delivery and operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.2	Fully defined external interfaces
	ADV_HLD.2	Security enforcing high-level design
	ADV_IMP.2	Implementation of the TSF
	ADV_LLD.1	Descriptive low-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	ALC_DVS.2	Sufficiency of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Test	ATE_COV.2	Analysis of coverage
	ATE_DPT.2	Testing: low-level design
	ATE_FUN.1	Functional test
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	AVA_MSU.2	Validation of analysis
	AVA_SOF.1	Strength of the TOE security function evaluation
	AVA_VLA.4	Highly resistant

5.2.1 Configuration Management (CM)

ACM_AUT.1 Partial CM automation

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

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

Evaluator action elements:

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

ACM_CAP.4 Generation support and acceptance procedures

Dependencies: ACM_SCP.1 TOE CM coverage
ALC_DVS.1 security measures 의 Identification

Developer action elements:

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

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

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

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

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

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

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

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

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

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items that comprise the TOE.

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

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

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

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

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

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

Evaluator action elements:

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

ACM_SCP.2 Problem tracking CM coverage

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

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

Content and presentation of evidence elements:

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

ACM_SCP.2.2C CM documentation shall describe how the Configuration items is tracked by the CM system.

Evaluator action elements:

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

5.2.2 Delivery and Operation

ADO_DEL.2 Detection of modification

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

Evaluator action elements:

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

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

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

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

Content and presentation of evidence elements:

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

Evaluator action elements:

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

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

5.2.3 Development

ADV_FSP.2 Fully defined external interfaces

Dependencies: ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

Evaluator action elements:

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

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

ADV_HLD.2 Security enforcing high-level design

Dependencies: ADV_FSP.1 Informal functional specification
ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

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

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

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

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

Evaluator action elements:

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

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

ADV_IMP.2 Implementation of the TSF

Dependencies: ADV_LLD.1 Descriptive low-level design
ADV_RCR.1 Informal correspondence demonstration
ALC_TAT.1 Well-defined development tools

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

Evaluator action elements:

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

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

ADV_LLD.1 Descriptive low-level design

Dependencies: ADV_HLD.2 Security enforcing high-level design
ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

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

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

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

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

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

Evaluator action elements:

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

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

ADV_RCR.1 Informal correspondence demonstration

Dependencies: No dependencies.

Developer action elements:

ADV_RCR1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

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

Evaluator action elements:

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

ADV_SPM.1 Informal TOE security policy model

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

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

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

Content and presentation of evidence elements:

ADV_SPM.1.1C The TSP model shall be informal.

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

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

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

Evaluator action elements:

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

5.2.4 Guidance Documents

AGD_ADM.1 Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

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

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

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

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

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

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

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

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

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

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

Evaluator action elements:

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

5.2.5 Life Cycle Support

ALC_DVS.2 Sufficiency of security measures

Dependencies: No dependencies.

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

Evaluator action elements:

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

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

ALC_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

ALC_TAT.1 Well-defined development tools

Dependencies: ADV_IMP.1 Subset of the implementation of the TSF

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

Evaluator action elements:

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

5.2.6 Test

ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.1 Informal functional specification
ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

ATE_DPT.2 Testing: low-level design

Dependencies: ADV_HLD.2 Security enforcing high-level design
ADV_LLD.1 Descriptive low-level design
ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.2.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.2.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.

Evaluator action elements:

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

ATE_FUN.1 Functional testing

Dependencies: No dependencies.

Developer action elements:

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

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

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

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

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

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

Evaluator action elements:

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

ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance
ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

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

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

5.2.7 Vulnerability Assessment

AVA_MSU.2 Validation of analysis

Dependencies: ADO_IGS.1 Installation, generation, and start-up procedures
ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

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

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 Strength of the TOE security function evaluation

Dependencies: ADV_FSP.1 Informal functional specification
ADV_HLD.1 Descriptive low-level design

Developer action elements:

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

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

AVA_VLA.4 Highly resistant

Dependencies: ADV_FSP.1 Informal functional specification
ADV_HLD.2 Security enforcing high-level design
ADV_IMP.1 Subset of the implementation of the TSF
ADV_LLD.1 Descriptive low-level design
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

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

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

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

Evaluator action elements:

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

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

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

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

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

6 TOE Summary Specification

6.1 TOE Security Functions

This section defines the 8 TOE Security Functions. Detailed information are classified thus put to blank in this public version of the Security Target.

6.2 Assurance Measures

[Table 4] TOE Assurance Measures

Assurance Class	Assurance component	Documentation
Configuration management	ACM_AUT.1	CM documents
	ACM_CAP.4	CM documents
	ACM_SCP.2	CM documents
Delivery and operation	ADO_DEL.2	Delivery documents
	ADO_IGS.1	Installation, generation, start-up documents
Development	ADV_FSP.2	Functional specification documents
	ADV_HLD.2	High-level design specification
	ADV_IMP.2	Implementation representation
	ADV_LLD.1	Detailed design specification
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Security policy model
Guidance documents	AGD_ADM.1	administrator Guidance documents
	AGD_USR.1	user Guidance documents
Life cycle support	ALC_DVS.2	Life cycle support documents
	ALC_TAT.1	Life cycle support documents
	ALC_LCD.1	Life cycle support documents
Tests	ATE_COV.2	Test documents
	ATE_DPT.2	Test documents
	ATE_FUN.1	Test documents
	ATE_IND.2	Test documents
Vulnerability assessment	AVA_MSU.2	Administrator Guidance documents, user Guidance documents
	AVA_SOF.1	Vulnerability analysis
	AVA_VLA.4	Vulnerability analysis

7 Protection Profile Claims

This section claims that the Security Target is in compliance with Smartcard Open Platform Protection Profile for Gevernment V1.0.

7.1 Protection Profile Reference

TOE satisfies all the requirements referenced from the following protection profile.

- Smartcard Open Platform Protection Profile for Gevernment V1.0 (2004. 12. 28)

7.2 Protection Profile Tailoring

None

7.3 Protection Profile Additions

The followings are additional security functional requirements to the security target. These refer to the Common Criteria.

- ADV_IMP.2 Implementation of the TSF
- ADV_DVS.2 Sufficiency of security measures
- ATE_DPT.2 Testing: low-level design
- AVA_VLA.4 Highly resistant

8 Security Objectives Rationale

This section describes rationale of security requirement, which satisfies security objectives, and security objectives which is defined based on security environment (threats, assumptions, organizational security policies). Rationale proves that the TOE provides efficient IT security measure in the context of the TOE security environment

8.1 Security Objectives Rationale

Rationale of security objectives proves that the specified security objectives are suitable, sufficient to handle security issues, and adequate without being excessive.

Security objective rationale proves the followings.

- Each assumption, threat, or organization's security policy is handled by at least one security objective.
- Each security objective handles at least one assumption, one threat, or one organizational security policy.

The following table shows that each security environment is mapped to at least one specific security objective.

[Table 5] Mapping of the security objectives to the security environments

security objectives security environment	TOE security objectives									security objectives for the environment							
	O.DAT_PROT	O.AUTH_ISS	O.AUTH_USR	O.APP_SEP	O.AUTH_RPR	O.ACC_AUTH	O.ROLLBACK	O.REM_RES	O.CNR_LEAK	OE.OPN_PLAT	OE.ATK_LEV	OE.TRAINING	OE.SEC_CNL	OE.APP_INS	OE.C_TAMPER	OE.UNDER_HW	OE.TSF_DAT
A.ATK_LEV											×						
A.SEC_CNL													×				
A.APP_INS														×			
A.UNDER_HW															×	×	
A.TOE_HNDL												×					
A.TSF_DAT																	×
T.LOGI_ATK	×	×	×		×	×											
T.ISS_ABU		×												×			
T.UNAU_IFD	×	×	×		×	×											

T.UNAU_APP			×			×											
T.SES_TEAR							×	×									
T.REP_AUTH						×											
T.ABN_END																×	
T.RES_COL				×													
T.RES_REU								×								×	
T.LEAK_INF									×								
T.TAMPER	×														×	×	
P.OPN_PLAT										×							
P.CRYPTO						×		×	×								
P.DUTY_SEP		×	×		×	×						×					

8.1.1 TOE Security Objectives Rationale

O.DAT_PROT

This security objective assures that only authorized user can access and modify user data. This security objective counters threats such as T.LOGI_ATK, that an unauthorized user induces attacks using smartcard commands, T.UNAU_IFD and T.TAMPER that physical attempts are conducted for unauthorized access to user data.

O.AUTH_ISS

This security objective assures that only authorized user can access and perform issuance function, when a smartcard is issued.

This security objective counters threats such as T.UNAU_IFD, that an unauthorized user conducts unauthorized access to IFD and issuance, T.ISS_ABU, and T.LOGI_ATK. This security objective also safisfies the organizational security policy, P.DUTY_SEP.

O.AUTH_USR

This security objective assures role identification of TOE user and TOE issuer, etc. TSF must clarify users who can use logical interface and assets that can be used. Hence, this security objective counters threats such as T.LOGI_ATK, T.UNAU_IFD, and T.UNAU_APP, and safisfies the organizational security policy, P.DUTY_SEP.

O,APP_SEP

This security objective assures that the TOE shall provide separation of each execution area for each application to avoid collision when sharing resources. Thus, this security objective counters T.RES_COL.

O.AUTH_RPR

This security objective assures that only an authorized issuer can access management function of smartcard, when the TOE comes to failure. This security objective counters T.LOGI_ATK, T.UNAU_IFD and safisfies P.DUTY_SEP

O.ACC_AUTH

This security objective assures that the TOE provides means of authentication to an identified user. Thus, this security objective counters T.LOGI_ATK, T.UNAU_IFD, T.UNAU_APP, and T.REP_AUTH, and safisfies P.CRYPTO and P.DUTY_SEP.

O.ROLLBACK

This security objective assures that the TOE resumes services from the previous valid state when detecting abnormality of the TSF service. Thus, this security objective counters T.SES_TEAR, that the TSF services are stopped by unintended failure.

O.REM_RES

This security objective assures leaving neither user data nor TSF data in work area used during TSF service. This counters threats such as T.RES_REU, that information is not removed properly from the resource after compleing usage including write operation, and T.SES_TEAR in case of abnormal end. This security objective also satisfies P.CRYPTO by removing cryptographic key information.

O.CNR_LEAK

This security objective assures countering abuse of the TSF data that may be leaked and caught by equipements even in case of normal operation. The TOE can be attacked in the environment where it is exposed physically and vulnerable. Thus, this security objective counters T.LEAK_INF. This security objective also satisfies P.CRYPTO by countering leakage of cryptographic key.

8.1.2 Security Objectives Rationale for the Environment

OE.OPN_PLAT

This security objective assures that the TOE carrying smartcard is an open platform which can load and operate multiple applications. And other components of smartcard must support operation of multiple applications as well as the TOE. Thus, this security objective satisfies P.OPN_PLAT.

OE.ATK_LEV

This security objective demonstrates to accept assumptions that the attacker has high level of domain knowledge, resources and motivation while finding out abusable vulnerabilities and abusing them with high probability. Thus, this security objective supports A.ATK_LEV.

OE.TRAINING

This security objective assures that training to each role shall be conducted in accordance with defined provisions from the manufacturing step to issuing and using step. The developer must specify adequate training in user maunal and administrator manual, and that shall be inspected by the evaluator. Thus, this security objective supports A.TOE_HNDL and P.DUTY_SEP.

OE.SEC_CNL

This security objective shall provide a secure channel between the TOE and IFD. Thus, this security objective supports A.SEC_CNL.

OE.APP_INS

This security objective assures that an installation of applications to be used in the TOE shall be conducted via approved procedures without fraud operation. Administrator shall install applications according to approved procedures including adequate identification and authentication, and the properly installed application shall not contain malicious code. Thus, this security objective supports A.APP_INS, and counters T.ISS_ABU.

OE.C_TAMPER

This security objective assures countering expert attackers' direct access to the IC or memory using chemical materials or elaborate equipments. Thus, this security objective counters T.TAMPER and support A.UNDER_HW assuming security of the underlying hardware.

OE.UNDER_HW

This security objective assures that the TOE operates in a physically secure chip, and the underlying hardware of the TOE shall have countermeasures against a variety of physical attacks. The TOE is an operating system placed in the smartcard chip that conducts resource management and application management, etc, hence the security of the underlying hardware is not able to be evaluated from the TOE design documentation and to be confirmed by the developer's or evaluator's vulnerability testing on it. The developer must use secure chip to achieve this security objective, and the evaluator shall verify it via testing and vulnerability analysis. Thus, this security objective supports A.UNDER_HW and counters T.ABN_END, T.RES_REU, and T.TAMPER.

.

OE.TSF_DAT

This security objective assures that the TSF data shall be managed securely even when it is in the environment out of direct control of the TOE. The TSF data stored in IFD must be managed securely to achieve this security objective. Thus, this security objective supports A .TSF_DAT.

8.2 Security Requirements Rationale

This section demonstrates that the combination of the security requirements is adequate to satisfy the identified security objectives.

[Table 6] Mapping of security functional requirements and security objectives

security objectives / Security Functional requirements	TOE security objectives								
	O.DAT_PROT	O.AUTH_ISS	O.AUTH_USR	O.APP_SEP	O.AUTH_RPR	O.ACC_AUTH	O.ROLLBACK	O.REM_RES	O.CNR_LEAK
FAU_ARP.1					×	×			
FAU_SAA.1					×	×			
FCS_CKM.1						×			×
FCS_CKM.4						×		×	
FCS_COP.1						×			×
FDP_ACC.2	×			×					
FDP_ACF.1	×			×					
FDP_RIP.1								×	
FIA_AFL.1		×			×	×			
FIA_ATD.1		×	×		×	×			
FIA_SOS.1						×			
FIA_UAU.1		×			×	×			
FIA_UAU.4		×			×	×			
FIA_UAU.6		×			×	×			
FIA_UID.1		×	×		×				
FMT_MOF.1	×								
FMT_MSA.1	×								
FMT_MSA.2	×								
FMT_MSA.3	×								
FMT_MTD.1		×							
FMT_MTD.2		×							
FMT_SMR.1		×	×		×	×			
FPT_AMT.1							×		
FPT_FLS.1							×		
FPT_PHP.3	×								
FPT_RCV.3							×		
FPT_RCV.4							×		

FPT_SEP.1				×					
FPT_TST.1							×		

8.2.1 TOE Security Requirements Rationale

TOE Securtity Requirements Rationale assures the following:

- Each TOE security objective is mapped to at least one TOE security functional requirements.
- Each TOE security requirement is mapped to at least on TOE security objective.

O.DAT_PROT

This security objective assures that only an authorized user can access user data. This security objective is met through the following security functional requirements of logical access control against logical threats and through FPT_PHP.3 including data deletion against physical threats.

FDP_ACC.2, FDP_ACF.1, FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FPT_PHP.3

O.AUTH_ISS

This security objective assures that only the authorized administrator of the TOE is able to have the role of loading, updating, deleting and managing application. This security objective is met through the following security functional requirements including role separation, identification and authentication of the user.

FIA_AFL.1, FIA_ATD.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.6, FIA_UID.1, FMT_MTD.1, FMT_MTD.2, FMT_SMR.1

O.AUTH_USR

This security objective assures identification of the TOE user role. This security objective is met through the following security functional requirements including identification function, role separation by user.

FIA_ATD.1, FIA_UID.1, FMT_SMR.1

O.APP_SEP

This security objective assures separation of each application area when multiple applications are operated. This security objective is met through the following security functional requirements of area separation.

FPT_ACC.2, FDP_ACF.1, FPT_SEP.1

O.AUTH_RPR

This security objective assures that only an authorized issuer can access and recover logical and physical properties of smartcard in case the TOE comes to failure that it can not be used. This security objective is met through the following security functional requirements including user's security alarm, clear role separation, identification and authentication, and security audit.

FAU_ARP.1, FAU_SAA.1, FIA_AFL.1, FIA_ATD.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.6, FIA_UID.1, FMT_SMR.1

O.ACC_AUTH

This security objective assures that the TOE provides proper means of authentication to an identified user who want to access data. This security objective is met through the following security functional requirements including cryptography support, authentication and role separation, security audit.

FAU_ARP.1, FAU_SAA.1, FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1 FIA_UAU.1, FIA_UAU.4, FIA_UAU.6, FMT_SMR.1

O.ROLLBACK

This security objective assures that the TOE shall keep well-defined valid state against failure and resume with them. This security objective is met through the following security functional requirements including the TOE's integrity check and automatic restoration.

FPT_AMT.1, FPT_FLS.1, FPT_RCV.3, FPT_RCV.4, FPT_TST.1

O.REM_RES

This security objective assures that user data or TSF data in work area are deleted to no unintended information remained in order for other subject to access them without authorization. This security objective is met through the following security functional requirements including cryptographic key destruction and remainder information deletion.

FCS_CKM.4, FDP_RIP.1

O.CNR_LEAK

This security objective assures that TSF data shall not be obtained by an attacker via analyzing electronical signal leakage when smartcard is used. This security objective can be implemented using hardware and software measures to counter leakage of information and shall be met through the following security functional requirements including cryptographic key generation and implementation of operations.

FCS_CKM.1, FCS_COP.1

8.2.2 TOE Assurance Requirements Rationale

The assurance level of this Security Target is EAL4+. This section describes the reason why EAL4 is selected and rationale with regard to components augmented in EAL4 assurance level.

[Table 7] Assurance requirements rationale

Assurance component	Basis/Security objectives
ACM_AUT.1	EAL4
ACM_CAP.4	EAL4

ACM_SCP.2	EAL4
ADO_DEL.2	EAL4
ADO_IGS.1	EAL4
ADV_FSP.2	EAL4
ADV_HLD.2	EAL4
ADV_IMP.2	O.CNR_LEAK, OE.OPN_PLAT
ADV_LLD.1	EAL4
ADV_RCR.1	EAL4
ADV_SPM.1	EAL4
AGD_ADM.1	EAL4
AGD_USR.1	EAL4
ALC_DVS.2	OE.TRAINING
ALC_LCD.1	EAL4
ALC_TAT.1	EAL4
ATE_COV.2	EAL4
ATE_DPT.2	O.CNR_LEAK, OE.OPN_PLAT
ATE_FUN.1	EAL4
ATE_IND.2	EAL4
AVA_MSU.2	EAL4
AVA_SOF.1	EAL4
AVA_VLA.4	O.CNR_LEAK, OE.UNDER_HW, OE.TSF_DAT OE.OPN_PLAT

Evaluation Assurance Level Rationale

EAL4 assurance requirement is an assurance package that requires systematic design, test and review. And EAL4 level is an assurance level of highest phase, that is required at commercial development step, and supply methodology that can be realized most definitely. Most smartcards are commercially developed and sold, and require high assurance level considering the value assets to be protected. It also provides higher assurance than EAL3 and, though partially, includes automated configuration management and requirement of secure distribution.

This assurance package selects higher assurance components than EAL4 partially. The followings describe the basis about augmented assurance componenets.

ADV_IMP.2 Implementation of the TSF

This Security Target selects ADV_IMP.2 instead of ADV_IMP.1 because it is necessary to confirm via source code that TSFs are implemented on software basis to counter attacks through TSF information leakage. And to achieve the TOE security objective such as O.CNR_LEAK and OE.OPN_PLAT, ADV_IMP.2, providing rather overall representation, is more suitable assurance component than ADV_IMP.1.

ALC_DVS.2 Sufficiency of security measures

This Security Target selects ALC_DVS.2 instead of ALC_DVS.1 because the evidence for development security must assure keeping confidentiality and integrity of the TOE in smartcard development. And to achieve TOE security objective OE.TRAINING, ALC_DVS.2, rather sufficient to provide security measures, is more suitable assurance component than ALC_DVS.1.

ATE_DPT.2 Testing: low-level design

EAL4 level requires detailed design (ADV_LLD.1) documents, but does not require test on them. However, module-wise testing is required in order to verify correct implementation and operation of the TOE, thus ATE_DPT.2, which requires module-wise testing, is augmented. Through module-wise testing, O.CNR_LEAK and OE.OPN_PLAT shall be achieved.

AVA_VLA.4 Highly resistant

Smartcard is possessed and used freely by individuals and there may be a possibility of high-level attacks such as those in university laboratory, hence AVA_VLA.4 is augmented to confirm that smartcard has high level of resistance against those attacks.

The TOE is an operatng platform that runs on smartcard IC chip and has resource management and security functions. Hence the TOE relies on the underlying smartcard IC chip in many ways including security and the security of the IC chip requires to be verified in order to assure security of not only the operating system, the TOE, but the whole of smartcard. AVA_VLA.4 is selected for this purpose and it verifies the capability of countering attacks against the TOE as well as the IC chip, partially.

8.3 Dependencies Rationale

8.3.1 Dependencies of the TOE Security Functional Requirement

The following table shows dependencies of the TOE security function components.

[Table 8] Dependencies of the TOE security function component

Index	Secrutity function component	Dependencies	Reference
1	FAU_ARP.1	FAU_SAA.1	2
2	FAU_SAA.1	FAU_GEN.1	None
3	FCS_CKM.1	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4 FMT_MSA.2	5 4 18
4	FCK_CKM.4	FDP_ITC.1 or FCS_CKM.1 FMT_MSA.2	3 18
5	FCS_COP.1	FDP_ITC.1 or FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	3 4 18
6	FDP_ACC.2	FDP_ACF.1	7

7	FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	6 19
8	FDP_RIP.1	-	-
9	FIA_AFL.1	FIA_UAU.1	12
10	FIA_ATD.1	-	-
11	FIA_SOS.1	-	-
12	FIA_UAU.1	FIA_UID.1	15
13	FIA_UAU.4	-	-
14	FIA_UAU.6	-	-
15	FIA_UID.1	-	-
16	FMT_MOF.1	FMT_SMR.1	22
17	FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1	6 22
18	FMT_MSA.2	ADV_SPM.1 FDP_ACC.1 or FDP_IFC.1 FMT_MSA.1 FMT_SMR.1	EAL4 6 17 22
19	FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	17 22
20	FMT_MTD.1	FMT_SMR.1	22
21	FMT_MTD.2	FMT_SMR.1	22
22	FMT_SMR.1	FIA_UID.1	15
23	FPT_AMT.1	-	-
24	FPT_FLS.1	ADV_SPM.1	EAL4
25	FPT_PHP.3	-	-
26	FPT_RCV.3	FPT_TST.1 AGD_ADM.1 ADV_SPM.1	29 EAL4 EAL4
27	FPT_RCV.4	ADV_SPM.1	EAL4
28	FPT_SEP.1	-	-
29	FPT_TST.1	FPT_AMT.1	23

8.3.2 Dependencies of Security Functional Requirements for IT Environment

FAU_GEN.1 dependent on FAU_SAA.1 is not satisfied. Smartcard does not have sufficient space to record security events, and an excessive security audit can cause hazard to the security of card accordingly. Thus, this Security Target does not include the requirement of FAU_GEN.1.

8.3.3 Dependencies of the TOE Assurance Requirement

The assurance package dependencies of EAL4 level provided by Common Criteria are already satisfied hence the rationale for them are omitted. The dependencies of augmented assurance requirements are as shown in [Table 9].

[Table 9] Dependencies of the augmented assurance components

index	Assurance component	Dependencies	Reference number

1	ADV_IMP.2	ADV_LLD.1 ADV_RCR.1 ALC_TAT.1	EAL4 EAL4 EAL4
2	ALC_DVS.2	None	-
3	ATE_DPT.2	ADV_HLD.2 ADV_LLD.1 ATE_FUN.1	EAL4 EAL4 EAL4
4	AVA_VLA.4	ADV_FSP.1 ADV_HLD.2 ADV_IMP.1 ADV_LLD.1 AGD_ADM.1 AGD_USR.1	EAL4 EAL4 1 EAL4 EAL4 EAL4

8.4 TOE Summary Specification Rationale

8.4.1 TOE Security Functions Rationale

The following table shows an overview, how the security functional requirements are satisfied by security functions.

[Table 10] Maping of security function and requirements

Security Function / Security funcional requiremnets	SF_Load	SF_Separation	SF_Transport	SF_Deletion	SF_Reuse	SF_Authentication	SF_Key	SF_Crypto
FAU_ARP.1	×		×	×			×	
FAU_SAA.1	×		×	×			×	
FCS_CKM.1								×
FCS_CKM.4								×
FCS_COP.1	×		×	×		×	×	×
FDP_ACC.2	×		×	×				
FDP_ACF.1	×							
FDP_RIP.1	×							
FIA_AFL.1	×							
FIA_ATD.1	×							×

FIA_SOS.1								×
FIA_UAU.1						×	×	
FIA_UAU.4							×	
FIA_UAU.6				×				
FIA_UID.1						×		
FMT_MOF.1	×					×		
FMT_MSA.1	×		×	×				
FMT_MSA.2	×		×	×				
FMT_MSA.3	×		×	×			×	
FMT_MTD.1						×		
FMT_MTD.2	×		×	×			×	
FMT_SMR.1	×		×	×		×		
FPT_AMT.1		×			×			
FPT_FLS.1						×		
FPT_PHP.3								×
FPT_RCV.3					×			×
FPT_RCV.4					×			
FPT_SEP.1		×						
FPT_TST.1		×			×			

- FAU_ARP.1, FAU_SAA.1

  SF_Load, SF_Transport, SF_Deletion, and SF_Key follow this SFR:

  TSF detects potential security violation and performs adequate reactions when loading, transferring, and deleting an application, and installating keys.

- FCS_CKM.1 , FCS_CKM.4

  SF_Crypto follows this SFR:

  This security function performs cryptographic operations provided by the TOE.

- FCS_COP.1

  SF_Load, SF_Transport, SF_Deletion, SF_Authentication, SF_Key, and SF_Crypto follow this SFR:

  This security function performs loading, transferring, and deleting an application, and performs authentication of smartcard, key installation, and cryptography control.

- FDP_ACC.2

  SF_Load, SF_Transport, and SF_Deletion follow this SFR:

  This security function manages access control of the TOE.

- FDP_ACF.1

  SF_Load follows this SFR:

  This security function manages access control which is based on the security attributes between subjects and objects.

- FDP_RIP.1

  SF_Load, SF_Reuse, and SF_Crypto follow this SFR:

  This security function handles protection of remainder information that retrieving resources back from objects, transferring buffer, etc.

- FIA_AFL.1

  SF_Load, SF_Deletion, and SF_Key follow this SFR:

  This security function sends error messages or disables the corresponding function in case the attempt count of authentication reaches or exceeds a defined limit.

- FIA_ATD.1

  SF_Load, SF_Separation, SF_Transport, and SF_Deletion follow this SFR:

  This security function keeps security attribute list belonging to each user.

- FIA_SOS.1

  SF_Crypto follows this SFR:

  This security function provides a mechanism to verify confidential information on RSA key according to the cryptography control of the TOE.

- FIA_UAU.1

  SF_Authentication and SF_Key follow this SFR:

  This security function checks if the MSM Controls Data be loaded before any application is loaded.

- FIA_UAU.4

  SF_Key follows this SFR:

  This security function provides a mechanism to prevent re-use when key is installed.

- FIA_UAU.6

  SF_Deletion follows this SFR:

  This security function provides re-authentication when deleting an application.

- FIA_UID.1

  SF_Authentication follows this SFR:

  This security function performs Check Data commands on behalf of a user before the user is identified.

- FMT_MOF.1

  SF_Load and SF_Authentication follow this SFR:

  This security function limits authorized role and authorized security functions.

- FMT_MSA.1, FMT_MSA.2

  SF_Load, SF_Transport and SF_Deletion follow this SFR:

  This security function provides management and secure value of security attributes according to application access control.

- FMT_MSA.3

  SF_Load, SF_Transport, SF_Deletion and SF_Key follow this SFR:

  This security function performs that the TOE shall have restricted value of security attributes used to force SFP.

- FMT_MTD.1

  SF_Authentication follows this SFR:

  This security function limits that only MCD issuer can request MULTOS Security Data.

- FMT_MTD.2

  SF_Load, SF_Transport, SF_Deletion and SF_Key follow this SFR:

  This security function manages the failure attempt number of times for loading application, transferring application, deleting application and key installation.

- FMT_SMR.1

  SF_Load, SF_Transport, SF_Deletion, and SF_Authentication follow this SFR:

  This security function conducts keeping the role of administrator or user of the TOE.

- FPT_AMT.1

  SF_Separation and SF_Reuse follow this SFR:

  This security function makes perform testing in case of start-up with repect to the AAM of the TOE.

- FPT_FLS.1

SF_Authentication and SF_Crypto follow this SFR:

This security function performs keeping secure state in case of data integrity failure, etc.

- FPT_PHP.3

  SF_Crypto follows this SFR:

  This security function provides resistance against electric power analysis to the TOE through outside interface.

- FPT_RCV.3, FPT_RCV.4

  SF_Reuse follows this SFR:

  This security function takes charge of automatic restoration and function restoration without resource loss against the case failure occurs or service stops.

- FPT_SEP.1

  SF_Separation follows this SFR:

  This security function handles separation of security function area between applications.

- FPT_TST.1

  SF_Separation and SF_Reuse follow this SFR:

  This security function provides function that verifies correct correct operation of the TSF and thie integrity of TSF data.

8.4.2 Strength of Security Function Level Rationale

The Strength of Security Function Level required by this Security Target is “SOF-high.”

Due to the definition of the TOE and the fact that a smartcard can be placed in a hostile environment, smartcard critical security mechanisms only have to be defeated by attackers possessing a high level of expertise, opportunity and resources, and successful attack is judged beyond normal practicality. The attacker is assumed to be thoroughly familiar with the specific implementation of the TOE and is presumed to have a high level of technical sophistication. Hence the TOE has countermeasure against them and it is very important that the claimed SOF should be high.

8.5 Protection Profile Claims Rationale

This section demonstrates that these rationales satisfy requirements of Smart Card Open Platform Protection Profile for Government V1.0(December 28, 1004).

8.6 Rationale about Mutual Supportiveness between Security Function Components

The purpose of this section is to analyze mutual supportiveness between security function components in terms that correct operation of the TOE security functions, integrity assurance of the TOE, and tolerance against failure are performed fiting well for the each own's purpose. This section focuses on four principal vulnerabilities and correspondent components of TOE security functions.

8.6.1 Bypass

Bypass means that non-permitted access to data or TSF by an unauthorized user. Bypass of security functions are prevented through the following functions.

FIA_UID.1, FIA_UAU.1, and FDP_RIP provide identification and authentication function to make only an authorized user can access to security functions.

FPT_PHP prevents bypass of security functions when the attacker access directly to the TOE data through physical attacks from outsisde.

8.6.2 Unauthorized Alteration

Unauthorized alteration of security functions are prevented by the following functions. The followings are security function components that assure integrity of the TOE and TSF data.

FPT_FLS.1, FPT_RCV.3, and FPT_RCV.4 assure secure states in case of failure of the TOE, thus unauthorized alteration shall be prevented.

FPT_PHP.3 prevents unauthorized alteration of the security functions caused by physical attacks from outside.

FPT_SEP.1 supports application separation function, protects execution areas specific to recources to prevent them from being altered without proper authorization.

FPT_TST.1 verifies data integrity, thus prevents unauthorized alteration of the TOE.

8.6.3 Disablement

Disablement of the security functions is prevented through the following functions.

FAU_ARP.1 provides security alarm function in case detecting attacks from outside, thus prevents security functionsof the TOE from being disabled.

FPT_PHP.3 detects physical attacks from outside, thus prevents security functions of the TOE from being disabled.

FPT_TST.1 verifies data integrity, thus prevents security functions of the TOE from being disabled.

8.6.4 Detection

Detection function is provided in case security functions are abused through the following functions.

FAU_SAA.1 performs monitoring for specified events, and analyzes to detect in case they reach threshold.

FIA_UID.1 and FIA_UAU.1 control access of the user who is not authorized for security alarm.

FMT_MTD.2 provides detecting and corresponding function in case TSF data exceeds specified level.

Annex

References

[1] Korea IT Security Evaluation and Certification Scheme, Ministry of Information and Communication, Korean Information Security Agency, September 22, 2005

[2] Korea IT Security Evaluation and Certification Guidance, Ministry of Information and Communication, Korean Information Security Agency, May 21, 2005

[3] Smartcard Handbook 3rd Edition, Wolfgang Rankl and Wolfgang Effing, John Wiley and Sons, Ltd., 2003.

[4] ISO/IEC 14443 Proximity Card

[5] ISO/IEC 7810 Identification Cards: Physical Characteristics

[6] ISO/IEC 7816 Identification Cards: Integrated Circuit Cards with Contacts

[7] ISO/IEC 10536 Close-coupled Cards

[8] IC card Protection Profile for Application Loading Mechanism (NMDA-ICC-PP), Japanese New Media Development Association, December 10, 2001

[9] Java Card^{TM} System Protection Profile Collection Version 1.0b, Sun Microsystems, August 2003

[10] Protection Profile Smartcard IC with Multi-Application Secure Platform Version 2.0, European Smartcard Industry Association, November 2000

[11] Smartcard Protection Profile Versiuon 3.0(SCSUG-SCPP), Smartcard Security User Group, September 2001

Abbreviations

CC	Common Criteria
CPU	Central Processing Unit
EAL	Evaluation Assurance Level
EEPROM	Electrically Erasable Programmable Read-Only Memory
IC	Integrated Circuit
RAM	Random Access Memory
ROM	Read-Only Memory
SFP	Security Function Policy
SOF	Strength of Function
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy

U-creator
U dream, we create.

END OF DOCUMENT

Phone: +82-2-3429-2114
E-mail: sdspr@samsung.com
http://sds.samsung.com

SAMSUNG SDS SAMSUNG