KEYCORP LIMITED

Keycorp MULTOS 1Q

ITSEC E6

Public Security Target

17 October 2003

Author: Keycorp MULTOS Team

Checked: _____

Document Number: SIM-SP-0204

Date: _____

Revision Number: 1.0

Contents

Preface

This document is an abridged and publicly available version of the Security Target that was submitted for the evaluation of MULTOS by LogicaCMG Australasia that lead to certification by DSD (certificate number 2003/30). Those parties wishing to obtain a copy of the entire Security Target should contact Keycorp Limited as follows:

MULTOS Technical Support
Level 5, Keycorp Tower
799 Pacific Highway
Chatswood NSW
Australia 2067

Telephone +612 9414 5200
Fax: +612 9415 1363
Email: multossupport@keycorp.net

1 Introduction

1.1 Background

1.1.1 The integrated circuit card (ICC), or smart card, is an ideal tool for the delivery of distributed, secure information processing at low cost. However, an application developed for one smart card is usually not portable to another. Furthermore, many current smart card operating systems allow only one application per card, meaning end users must carry a multitude of cards, one for each function or service required.

1.1.2 Keycorp Ltd, in its role as a member of the MAOSCO consortium, is developing an open, high-security multi-application operating system to address the current shortcomings of smart card operating systems. This operating system is called MULTOS.

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

a) execute an application written for MULTOS - application execution should be independent of the underlying smart card hardware

b) load many applications - applications should be able to co-exist on the smart card

c) ensure that applications are securely loaded and segregated - they should not be able to interfere with each other or with MULTOS.

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

1.2 Purpose

1.2.1 Keycorp is submitting MULTOS masked on Infineon Technologies SLE66CX160P and SLE66CX320P Smartcard Integrated Circuits as the Target of Evaluation (TOE) of a formal security evaluation. MULTOS is to be evaluated against the Information Technology Security Evaluation Criteria (ITSEC, reference 1). The evaluation will be performed by LogicaCMG in

Australia, under the Australasian ITSEC evaluation scheme. This document, the MULTOS Security Target, forms the baseline for such an evaluation. It provides a complete and consistent statement of the security enforcing functions and mechanisms of the TOE.

1.3 Target of Evaluation

- The Target of Evaluation is Keycorp MULTOS version 1Q, including AMD with ID (as assigned by MULTOS CA) 0020v003, masked on Infineon Technologies SLE66CX160P and SLE66CX320P Smartcard Integrated Circuits. Keycorp MULTOS implements version 4.06 of the MULTOS specifications.

The target of evaluation therefore includes both the MULTOS operating system software and the SLE66CX160P and SLE66CX320P Smartcard Integrated Circuits that it is masked on.

1.4 Scope

1.4.1 The Security Target details the TOE's security objectives and the security enforcing functions proposed to address the assumed threats to the assets protected by the TOE.

1.4.2 This Security Target is structured as follows:

a) Chapter 1 provides an overview of the document and background details on MULTOS; the minimum strength of mechanisms and the target evaluation level are stated

b) Chapter 2 is a product rationale; the product's security objectives and a summary of its security features is presented, along with the intended method of use, the intended environment and the assumed threats within that environment

c) Chapter 3 is a specification of the security enforcing functions; these functions are countermeasures that are devised to address the assumed threats to the product in its intended environment.

1.5 Claimed Rating of Minimum Strength of Mechanisms

1.5.1 The Strength of Mechanisms Analysis for MULTOS (to be developed as part of the MULTOS Effectiveness Documentation) will document the analysis of the strength of the weakest critical Type A mechanism in MULTOS.

1.5.2 The claimed rating of the minimum strength of mechanisms in MULTOS is high.

1.6 Target Evaluation Level

MULTOS is intended to meet the ITSEC E6 evaluation level.

1.7 References

1) Information Technology Security Evaluation Criteria, Commission of the European Communities, version 1.2, 28 June 1991.

2) ISO/IEC 7816: Identification Cards - Integrated Circuit(s) Cards with Contacts.

2 Product Rationale

2.1 Product Description

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

2.1.2 The user of the smart card accesses the applications loaded on it via an Interface Device (IFD), which could be a Point-of-Sale terminal, Automatic Teller Machine, or some other device which supports ISO 7816 smart card protocols (see reference 2).

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

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

a) applications to be loaded onto and deleted from the smart card.

b) an IFD to access data and applications which are loaded on the card.

c) information specific to the card to be retrieved by an IFD.

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

a) wait for input from the IFD.

b) parse the input.

c) if the input is a MULTOS command, process the command and write a response to the IFD.

d) otherwise, execute the currently selected application and write to the IFD any output created by the application.

e) loop back to a).

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

2.1.7 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 smart card personalisation 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.

2.2 Security Objectives

MULTOS is intended to achieve the following security objectives:

a) to preserve the mutual confidentiality of multiple applications loaded and executed on a single smart card (Conf_Obj_1)[1].

b) to preserve the mutual integrity of multiple applications loaded and executed on a single smart card (Int_Obj_1).

c) to confirm the authority of all application load and delete requests (Auth_Obj_1).

2.3 Intended Method of Use

2.3.1 MULTOS is intended to provide a hardware-independent environment for the execution of multiple applications which provide a variety of functions and services to the holder of the smart card. Applications may be developed and supplied by different organisations 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).

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

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

2.3.4 MULTOS provides the following features.

U1 MULTOS will ensure all requests to load applications are appropriately authorised. MULTOS will support a capability to ensure the authenticity and integrity of an application when loading the application onto the smart card. MULTOS will also ensure all requests to delete applications are appropriately authorised. Reasons for wishing to delete applications may be because they are found to contain errors, because an updated application is available, or to make room on the smart card for a more desirable application.

U2 MULTOS will support a capability to load encrypted applications onto the smart card, decrypt such applications and make them available to the smart card user for execution.

U3 MULTOS will ensure no application loaded on the smart card can interfere with the operation of any other loaded application or with MULTOS. MULTOS will also ensure that an application's code and data will not be available to other applications after it has been deleted.

U4 MULTOS will provide the capability to authenticate a card as a valid MULTOS-equipped smart card.

U5 MULTOS will provide the capability to restrict the use of regulated features of the smart card (e.g., strong cryptography) to authorised applications.

[1] The notation used here represents the key characteristic of the security objective (e.g., Confidentiality, Integrity, Authorisation). It is used to assist traceability throughout the Security Target.

U6 MULTOS defines certain functions (installing keys, loading applications and deleting applications) as sensitive functions. For each of these functions, if the number of failed attempts to execute the function reaches a pre-defined limit over the life of the smart card, MULTOS will permanently disable the function. In the case of installing keys, this means the card is unusable, as no applications can be loaded until keys have been installed. In the cases of application loading and deleting, other functions of the card remain available.

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

2.4 Intended Environment

2.4.1 TOE Location and Usage

2.4.1.1 The MULTOS operating system will initially be developed in software. Following successful implementation and testing, the MULTOS executable will be masked in Read Only Memory (ROM) and embedded on smart cards.

2.4.1.2 Once the MULTOS chip has been embedded on a target smart card, interaction with it will be via commands issued to the card from an IFD or service requests (i.e., MULTOS system calls, known as primitives) made by an executing application.

2.4.2 Hardware and Firmware

2.4.2.1 This implementation operates on Infineon Technologies SLE66CX160P and SLE66CX320P Smartcard Integrated Circuits (ICs). The IC provides the microprocessor to execute the instructions comprising the executable code of MULTOS. It also includes hardware support for the implementation of cryptographic functions. This support is in the form of 512 and 1024 bit registers and associated instructions to manipulate data in these registers. MULTOS requires the target IC to execute instructions correctly according to its specification. No other protection mechanisms are required of the target hardware.

2.4.2.2 MULTOS requires firmware run-time libraries to support writing data to EEPROM. These libraries are supplied by Infineon Technologies. 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 the run-time libraries to execute correctly according to specification, to ensure data is written to the correct address within EEPROM.

2.4.2.3 MULTOS also requires firmware run-time libraries to support manipulation of 512 and 1024 bit registers. These in turn support the cryptographic functions provided by MULTOS. MULTOS requires these run-time libraries to execute correctly according to specification to ensure correct manipulation of the 512 and 1024 bit registers occurs.

2.4.3 Supporting Security Infrastructure

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

2.4.3.2 The following roles and responsibilities are assumed within the infrastructure:

a) **MULTOS Security Manager (MSM):** defines and polices the MULTOS security infrastructure and provides criteria and services necessary for MULTOS participants to operate within the infrastructure. It acts as Certification Authority for the security infrastructure. It is assumed only one MSM exists. The MSM must be trusted by all participants in the infrastructure.

b) **MULTOS Implementor:** the organisation that implements a MULTOS version. The MULTOS Implementor is licensed by MAOSCO and provides its MULTOS version to the IC Manufacturer. The MULTOS Implementor requests the MSM to provide MSM Controls Data, although this may be delivered to the MCD Manufacturer or MCD Issuer.

c) **Integrated Circuit (IC) Manufacturer:** manufacturer of silicon from which chips and smart cards are made. It is assumed the IC Manufacturer is trusted to perform its tasks correctly. It is assumed that all security measures concerned with card manufacture are completed at the IC manufacture stage. This includes:

i) the inclusion of security keys in the ROM mask
ii) the injection of security data into non-volatile memory.

Security keys and data are provided by the MSM. The initialised ICs are provided to MCD Manufacturers.

d) **MULTOS Carrier Device (MCD)[2] Manufacturer:** responsible for embedding the IC in its plastic carrier and for background printing on the card. The result is an initialised MCD. This operation is assumed not to be security sensitive. The MCD Manufacturer may also receive MSM Controls Data from the MSM and enable the MCDs[3]. Initialised and enabled MCDs are provided to MCD Issuers.

e) **MCD Issuer:** responsible for issuing to users the MCD itself. The MCD Issuer may also enable initialised MCDs, by loading MSM Controls Data received from the MSM onto the MCDs. MCD Issuers retain the ultimate authority over what applications are loaded on their MCDs. MCD Issuers register applications with the MSM, provide information related to the applications and receive application load and delete certificates from the MSM.

f) **Application Writer:** licensed by MAOSCO to produce applications for MULTOS. Supplies applications under contract to Application Issuers.

g) **Application Issuer:** an organisation which wishes to offer 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.

[2] MULTOS Carrier Device (MCD) is the term used for a smart card on which MULTOS has been installed. It will be used throughout the remainder of this document.
[3] When first manufactured, a MCD is termed “initialised”. After MSM Controls Data is installed, it is termed “enabled”. Throughout this document, “MCD” is understood to mean “enabled MCD” unless otherwise stated.

h) **Application Provider:** the organisation which takes responsibility for an application, by certifying it with the organisation's public key and encrypting it where necessary. The Application Provider is a role that can be performed by an Application Writer, Application Issuer or MCD Issuer, rather than necessarily, being an organisation in its own right.

i) **Application Loader:** responsible for performing the technical operation of loading applications onto MCDs. The Application Loader enters into an agreement with one or more Application Issuers and MCD Issuers for loading applications supplied by one or more Application Providers.

j) **MCD User:** final user of the MCD.

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

2.4.3.4 MA-cards are assumed to satisfy the following requirements:

a) they are manufactured in a controlled environment conforming to MSM rules.

b) they are subject to type approval by the MSM.

c) they possess a level of tamper resistance.

d) they shall be initialised, enabled and issued as MCDs according to MSM rules.

2.4.4 Application Load Units

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

2.4.5 Key Transformation Unit

An Application Provider wishing to utilise application confidentiality will generate a Key Transformation Unit (KTU). The KTU contains descriptors for the areas of the application's code and data that have been encrypted. Each descriptor contains the start address of the protected area, the length of the protected area, an indicator of the algorithm used and the key used to encrypt the contents of the area. The descriptors and some header information (including application identifier and target MCD number) are then encrypted, using the target MCD's public transport key (see section 2.4.6), and included in the KTU.

2.4.6 Application Load and Delete Certificates

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

a) a specific MCD.

b) a subset of the cards issued by a MCD Issuer.

c) all cards issued by a MCD Issuer.

d) limited to a subset of cards enabled on specific dates.

e) a combination of the above.

2.4.6.2 An ALC contains load controls which define exactly what load operations are allowed. The load controls specify:

a) if application certification has been used.

b) if application confidentiality has been used.

c) if reloading a deleted application is permitted.

2.4.6.3 The ALC also contains feature permissions which define what regulated features the application may use. For the initial version of MULTOS, the only regulated features are strong cryptography functions.

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

2.5 Assumed Threats

2.5.1 The threats identified in this section are assumed to exist in the environment in which MULTOS will operate.

2.5.2 **T1: Unauthorised load of an application onto a smart card.** An unauthorised load of an application onto a smart card is performed, leading to loss of revenue for card suppliers or application providers. This is a threat to security objective Auth_Obj_1.

2.5.3 **T2: Unauthorised modification of an application prior to loading onto a smart card.** The integrity of an application may become compromised before it has been loaded onto a smart card. This is a threat to security objective Auth_Obj_1.

2.5.4 **T3: Unauthorised reload of an application onto a smart card.** Applications may contain initial value (e.g., electronic purse applications). Such an application could be deleted after the value has been consumed and a copy of the load sequence replayed in order to create value without authorisation. This is a threat to security objective Auth_Obj_1.

2.5.5 **T4: Application is loaded onto an unauthorised smart card.** An unauthorised smart card cannot be trusted to execute applications correctly or to maintain the security of information held by the smart card. An attacker who is able to create an unauthorised smart card could use it to attack applications and their data. This is a threat to security objective Conf_Obj_1.

2.5.6 **T5: Application executing on a smart card compromises the security of the smart card or of another application on the smart card.** This threat could be deliberate or accidental. An attacker who is able to gain authorisation for an application could use the application to attack the smart card operating system or other applications found on the smart card. A legitimate application provider could unintentionally develop an application which, due to implementation error, interferes with the operation of the smart card operating system or other applications on the smart card. This is a threat to security objectives Conf_Obj_1 and Int_Obj_1.

2.5.7 **T6: Application is deleted from a smart card without authorisation.** An attacker could direct the smart card to delete an application from the smart card, preventing legitimate users from accessing its functionality or causing a widespread loss of monetary value. This is a threat to security objective Auth_Obj_1.

2.5.8 **T7: Unauthorised access is gained to sensitive information held within a smart card application.** A smart card application may contain sensitive data. This could be compromised before the application has been loaded onto the smart card by attacking the communication channel used to load the application. Sensitive data could also be compromised after the application has been deleted from the smart card, by another application loaded into the deleted application's process and memory space. This is a threat to security objective Conf_Obj_1.

2.5.9 **T8: Unauthorised use of strong cryptography.** An application which is not authorised to do so accesses strong cryptography services provided by the smart card. This is a threat to security objective Auth_Obj_1.

2.6 Summary of Security Features

2.6.1 MULTOS provides the following security functionality:

a) application separation

b) application load and authentication

c) application access control

d) application confidentiality

e) application deletion with object reuse

f) smart card authentication

g) control of regulated smart card features.

2.6.2 MULTOS will maintain separate data and execution environments for all loaded applications to ensure no application can interfere with the operation of another.

2.6.3 MULTOS will ensure all application load requests are authorised by the MSM prior to loading applications onto the MCD. MULTOS will optionally authenticate the application and verify its integrity.

2.6.4 MULTOS will check the permissions associated with an application to ensure the application is permitted to be loaded onto the MCD.

2.6.5 MULTOS will provide the capability to load and execute applications which have encrypted components. An application developer may choose to encrypt application code or data in order to maintain its confidentiality prior to loading on to the MCD. Only authentic MCDs will be capable of loading protected applications.

2.6.6 MULTOS will ensure all application delete requests are authorised by the MSM prior to deleting applications from the MCD. MULTOS will ensure that an application's code and data will not be available to other applications after it has been deleted.

2.6.7 MULTOS will provide the capability to authenticate a card as a valid initialised MCD prior to the card being enabled.

2.6.8 MULTOS will provide the capability to restrict the use of regulated features of the smart card (e.g., strong cryptography) to authorised applications.

3 Specification of Security Enforcing Functions

3.1 Introduction

3.1.1 In this chapter, each Security Enforcing Function (SEF) of MULTOS is specified. The following format is used to present each of the SEFs:

- Overview
- Specification
- Threats Addressed.

3.1.2 The Overview provides a brief description of the SEF, correlating it to the intended method of use of MULTOS and the assumptions regarding its operational environment defined in the Product Rationale (Chapter 2).

3.1.3 The Specification provides a statement of the security functionality provided by the SEF.

3.1.4 Threats Addressed identifies which of the assumed threats identified in the Product Rationale are addressed by the SEF. An explanation is provided of how the SEF is adequate to counter the assumed threats it addresses.

3.2 Application Load and Authentication SEF

3.2.1 Overview

3.2.1.1 The Application Load and Authentication SEF correlates to Methods of Use U1 and U6 and directly supports the Auth_Obj_1 security objective.

3.2.1.2 An Application Loader is able to submit a series of commands to MULTOS, via an IFD, to load the contents of an Application Load Unit (ALU) onto the MCD. Included in the information submitted in the load commands is the Application Load Certificate (ALC). This is a public key certificate, signed by the MSM, which certifies the application load request has been authorised by the MSM.

3.2.1.3 The ALC also contains details of the load permissions, load controls and feature permissions associated with the application. One of the load controls specifies if the Application Provider has signed the application to authenticate it. If so, the application's certificate is also submitted to MULTOS. MULTOS uses this to ensure the authenticity and integrity of the application.

3.2.1.4 An application load attempt will fail if the MCD has insufficient memory.

3.2.1.5 Application Load and Authentication is regarded as a sensitive operation, so MULTOS strictly limits the number of unsuccessful attempts to perform the operation that can be made in the life of the MCD.

3.2.2 Specification

3.2.2.1 MULTOS shall authenticate an application load request as having been authorised by the MSM, prior to loading the application.

3.2.2.2 MULTOS shall ensure an authorised application has appropriate permissions before it is loaded.

3.2.2.3 Only one copy of an application can be loaded on a MCD at any time.

3.2.2.4 Application authentication is optional. When invoked, the process of authentication shall verify the authenticity of the application.

3.2.2.5 The process of authentication shall verify the integrity of the authorised application.

3.2.2.6 MULTOS shall not reload an application which has been previously loaded, then deleted, unless authorised by the MSM.

3.2.2.7 MULTOS shall not load any application unless MSM Controls Data has first been loaded.

3.2.2.8 Application Load and Authentication is a sensitive function. MULTOS shall keep a count of the number of failed attempts to execute this function. If this count reaches a defined limit, MULTOS shall permanently disable this function.

3.2.3 Threats Addressed

3.2.3.1 The Application Load and Authentication SEF addresses the following threats:

- T1: Unauthorised load of an application onto a smart card
- T2: Unauthorised modification of an application prior to loading onto a smart card
- T3: Unauthorised reload of an application onto a smart card.

3.3 Application Separation SEF

3.3.1 Overview

3.3.1.1 The Application Separation SEF correlates to Method of Use U3 and directly supports the Conf_Obj_1 and Int_Obj_1 security objectives.

3.3.1.2 The Application Separation SEF ensures each application is restricted to accessing its own code and data, except as noted below. Any attempt by the application to directly execute code outside its own code space, or access data outside its own data space is blocked by MULTOS and the application is aborted. In this way, MULTOS ensures applications cannot interfere with each other or with MULTOS.

3.3.1.3 The only exceptions to the restriction on an application's code and data access are as follows:

a) accessing data in the Public data area

b) application delegation

c) accessing Codelets

d) accessing data via MULTOS primitives.

3.3.1.4 An application can make data available to other applications or to the outside world by writing the data to the Public area. The Public area is included in the data address space of every application loaded on the MCD. Therefore, an application can always read the contents of the Public area, which may contain data placed there by another application. In addition, MULTOS uses the Public area as its input/output buffer for communication with the IFD. Therefore, an application can write data into the Public area and MULTOS will write it to the IFD.

3.3.1.5 An application can delegate its execution to another application. When an application delegates, its execution state is suspended by MULTOS and the delegated application commences execution. When the delegated application finishes its execution, the delegator resumes execution at the point where it was suspended. A delegated application can itself delegate to another application, but recursive delegation (either directly or indirectly) is not permitted. In this way, applications can co-operate to perform a particular task. Delegation is also the mechanism for allowing one application to read the data of another application via the Public data area.

3.3.1.6 An application can transfer execution to a shared code routine (known as a Codelet). This mechanism is similar to a subroutine or procedure call. The Codelet has its own code space for execution, but executes in the calling application's data space. When the Codelet finishes execution, control is returned to the calling application. Codelets are controlled by MULTOS and can be loaded onto the MCD when it is initialised or enabled.

3.3.1.7 An application can obtain information about the MCD on which it is loaded and on the state of MULTOS by calling MULTOS primitives. MULTOS returns the requested data to the calling application. MULTOS also provides primitives that allow applications to update the ATR file. This is the only system data an application can update.

3.3.2 Specification

3.3.2.1 MULTOS shall maintain separate storage and execution space for applications loaded onto a MCD.

3.3.2.2 An application shall be able to read code for execution only from its own code space or from a pool of common routines controlled by MULTOS.

3.3.2.3 No application shall be able to write to the code space of any application, including itself.

3.3.2.4 No application shall be able to read from or write to the data space of another application except via a mechanism provided and controlled by MULTOS.

3.3.2.5 No application shall be able to cause the execution of another application except via a mechanism provided and controlled by MULTOS.

3.3.2.6 No application shall be able to write to the code space of MULTOS.

3.3.2.7 No application shall be able to read from or write to the data space of MULTOS except via a mechanism provided and controlled by MULTOS.

3.3.3 Threats Addressed

3.3.3.1 The Application Separation SEF addresses the following threat:

- T5: Application executing on a smart card compromises the security of the smart card or of another application on the smart card.

3.4 Application Transport Confidentiality SEF

3.4.1 Overview

3.4.1.1 The Application Transport Confidentiality SEF correlates to Methods of Use U2 and U6 and supports the Conf_Obj_1 security objective.

3.4.1.2 An Application Provider is able to protect the confidentiality of its applications prior to the applications being loaded onto a MCD. The Application Provider creates a KTU. The ALC indicates that application confidentiality is being used. The KTU is loaded onto the MCD as part of the application load process. MULTOS decrypts the KTU and uses the information contained within it to decrypt the protected portions of the loaded application. This ensures the confidentiality of the application until it has been successfully loaded. It also verifies the authenticity of the MCD, since only the target MCD for the application will be able to successfully decrypt the KTU.

3.4.1.3 Application Transport Confidentiality is regarded as a sensitive operation, so MULTOS strictly limits the number of unsuccessful attempts to perform the operation that can be made in the life of the MCD.

3.4.2 Specification

3.4.2.1 MULTOS shall be able to load authorised applications which have protected areas of code or data.

3.4.2.2 MULTOS shall be able to remove the protection once the application is loaded and shall be able to execute the application.

3.4.2.3 Only an authentic MCD shall be able to load and execute a protected application.

3.4.2.4 Application Transport Confidentiality is a sensitive function. MULTOS shall keep a count of the number of failed attempts to execute this function. If this count reaches a defined limit, MULTOS shall permanently disable this function.

3.4.3 Threats Addressed

3.4.3.1 The Application Transport Confidentiality SEF addresses the following threats:

- T4: Application is loaded onto an unauthorised smart card
- T7: Unauthorised access is gained to sensitive information held within a smart card application.

3.5 Application Deletion SEF

3.5.1 Overview

3.5.1.1 The Application Deletion SEF correlates to Methods of Use U1 and U6 and directly supports the Auth_Obj_1 security objective.

3.5.1.2 The Application Deletion SEF ensures requests to delete applications have been appropriately authorised. The MSM authorises a delete request by creating an Application Delete Certificate (ADC) for the application. To delete an application, the ADC is submitted to the MCD as the parameter of a Delete Application command issued to MULTOS via an IFD. MULTOS verifies the authenticity of the ADC and ensures the information it contains matches a loaded application. This confirms the delete request has been authorised by the MSM and is for the correct application.

3.5.1.3 Application Deletion is regarded as a sensitive operation, so MULTOS strictly limits the number of unsuccessful attempts to perform the operation that can be made in the life of the MCD.

3.5.2 Specification

3.5.2.1 MULTOS shall authenticate all requests to delete applications.

3.5.2.2 The process of authentication shall verify that the delete request has been authorised by the MSM.

3.5.2.3 MULTOS shall delete an application only after receiving and authenticating a valid application delete request.

3.5.2.4 Application Deletion is a sensitive function. MULTOS shall keep a count of the number of failed attempts to execute this function. If this count reaches a defined limit, MULTOS shall permanently disable this function.

3.5.3 Threats Addressed

3.5.3.1 The Application Deletion SEF addresses the following threat:

- T6: Application is deleted from a smart card without authorisation.

3.5.3.2 An application can only be deleted by providing MULTOS with an ADC. The ADC must contain the correct AID and permissions and the same random number as the original ALC. The ADC is digitally signed by the MSM using the secret GKCK. Therefore, MULTOS is assured that the delete request has been authorised by the MSM.

3.6 Object Reuse SEF

3.6.1 Overview

3.6.1.1 The Object Reuse SEF correlates to Method of Use U3 and directly supports the Conf_Obj_1 security objective.

3.6.1.2 The Object Reuse SEF ensures that, once an application is deleted from a MCD, no trace of its code or data remains on the MCD (with the exception of any data written to the Public data area). This is done by overwriting all of the application's code and data spaces with fixed data. In this way, any application subsequently loaded into the same memory area will be unable to determine any information related to previously loaded applications.

3.6.1.3 Data in the Public data area is not subject to object reuse, as the Public area provides the mechanism for applications to share data. It is assumed any data written by an application to the Public data area is not sensitive and is intended to be accessed by other applications or by MULTOS for output to the IFD.

3.6.2 Specification

MULTOS shall ensure that no part of an application's code or data, excluding data the application has placed into the Public data area, can be accessed after the application has been deleted.

3.6.3 Description and Explanation

3.6.3.1 When MULTOS deletes an application from the MCD, it shall overwrite the application's code and data spaces with a fixed pattern of bytes. In this way, any other application subsequently loaded into the same space will be unable to determine any information relating to the deleted application.

3.6.3.2 Data which the application has written to the Public data area is not overwritten, since this provides the means for the application to communicate with other applications. By placing data in the Public data area, an application is effectively deciding the data can be accessed by any application.

3.6.4 Threats Addressed

3.6.4.1 The Object Reuse SEF addresses the following threat:

- T7: Unauthorised access is gained to sensitive information held within a smart card application.

3.6.4.2 Ignoring physical attack of the MCD (which is beyond the scope of evaluation), unauthorised access to sensitive information held within an application could be achieved by:

a) intercepting the application before it is loaded onto the MCD - this is addressed by the Application Confidentiality SEF (see section 3.4)

b) loading an application onto the MCD and reading the code or data space of the target application - this is addressed by the Application Separation SEF (see section 3.3)

c) loading an application onto the MCD and recovering information of a previously loaded and deleted application from the data space - MULTOS addresses this by overwriting code and data space when an application is deleted.

3.7 Smart Card Authentication SEF

3.7.1 Overview

3.7.1.1 The Smart Card Authentication SEF correlates to Method of Use U4 and supports the Key Installation SEF in meeting the MULTOS security objectives.

3.7.1.2 The Smart Card Authentication SEF provides a means for MCD Issuers to determine that a MCD is an authentic initialised MCD prior to loading it with MSM Controls Data. This is done using the Check Data Command. The Check Data Command creates a hash digest over a specifed memory area. It includes in the digest fixed MULTOS data and a random challenge issued as part of the Check Data Command. The resultant digest can be compared with one produced by exactly the same command issued to a known authentic initalised MCD. If the digests are the same, the MCD Issuer can be guaranteed that the MCD being checked is also an authentic initialised MCD.

3.7.2 Specification

3.7.2.1 On request, MULTOS shall provide a digest of the contents of a selected area of memory within an initialised MCD.

3.7.2.2 The digest shall incorporate a portion of fixed MULTOS data.

3.7.2.3 The digest shall be representative of the contents of the memory which is subject to authentication (i.e., the selected area of memory together with the fixed portion of MULTOS data).

3.7.2.4 It shall not be possible to infer from the digest any information regarding the contents of the memory area checked.

3.7.2.5 This functionality shall be available only on initialised MCDs (i.e., MCDs which have not yet been enabled).

3.7.3 Description and Explanation

On request, MULTOS shall provide a digest of the contents of a selected area of memory within an initialised MCD.

3.7.3.1 MULTOS provides a facility, known as the Check Data Command, to generate a digest of a specified area of memory on the MCD. This can be used for comparison with the results of the same Check Data Command applied to a known authentic initialised MCD, in order to verify the authenticity of the target MCD.

The digest shall incorporate a portion of fixed MULTOS data.

3.7.3.2 The Check Data Command requires as input:

a) the start address of the memory area to be checked

b) the length of the memory area to be checked

c) a random challenge value.

3.7.3.3 MULTOS performs a bit-wise exclusive OR function on the random challenge value and the first part of the fixed transport key (tkf, see Table 2-1). The result of this operation is concatenated with the second part of tkf and a one way hash algorithm applied to it. Using this hash as an initial value, a hash digest is computed over the contents of the indicated memory area.

3.7.3.4 The inclusion in the digest of fixed MULTOS data (in the form of tkf) enables the authenticity of the MCD to be checked by comparing the digest with the result produced from a Check Data Command applied to the same area of memory on a known authentic initialised MCD.

The digest shall be representative of the contents of the memory which is subject to authentication (i.e., the selected area of memory together with the fixed portion of MULTOS data).

3.7.3.5 A one-way hash function is applied to the fixed data, random challenge value and memory area to be checked. This also contributes to checking the authenticity of the MCD. The random challenge value ensures the returned digest cannot be spoofed.

It shall not be possible to infer from the digest any information regarding the contents of the memory area checked.

3.7.3.6 The digest is formed using a one-way hash function over the specified memory area and random challenge value. This acts to prevent any useful information being returned in the digest and therefore prevents any potential compromise of sensitive MULTOS information.

This functionality shall be available only on initialised MCDs (i.e., which have not yet been enabled).

3.7.3.7 The Check Data Command is only useful for authenticating MCDs before they are enabled. Allowing its use after the MCD is enabled could provide a means for probing for information related to applications loaded on the MCD. As it serves no useful purpose once the MCD is enabled and, despite the way in which the digest is constructed, could be used to attack the MCD, it is prudent to disable its functionality after loading MSM Controls Data.

3.7.3.8 If the Check Data Command is applied to an enabled MCD, MULTOS returns an error condition. No digest is calculated.

3.7.4 Security Mechanisms

The Smart Card Authentication SEF shall use the following security mechanism:

a) a one-way hash function, to determine the integrity of memory within an MCD.

3.7.5 Threats Addressed

3.7.5.1 The Smart Card Authentication SEF addresses the following threat:

- T4: Application is loaded onto an unauthorised smart card.

3.7.5.2 This threat is addressed by the ability to obtain a digest of a specific area of memory on an initialised MCD. It also relies on having a known authentic MCD and being able to apply the same random challenge to both the known MCD and the MCD to be checked. If the returned digests are the same, the authenticity of the challenged MCD can be assumed. If the returned digests are different, it can be assumed either an unauthorised change has been made to the area of memory checked on the challenged MCD, or the card is not an authentic MCD.

3.8 Key Installation SEF

3.8.1 Overview

3.8.1.1 The Key Installation SEF correlates to Methods of Use U1, U2 and U6. It supports the Application Load and Authentication SEF and Application Transport Confidentiality SEF in meeting the MULTOS security objectives.

3.8.1.2 An initialised MCD must be enabled with its permissions and transport key set before any applications can be loaded onto it. MULTOS provides the Set MSM Controls command to allow MCDs to be enabled. MSM Controls Data is encrypted using a MCD-specific symmetric key (tkv, see Table 2-1). The encrypted data is then submitted to MULTOS as part of the Set MSM Controls command. MULTOS decrypts the MSM Controls Data and verifies that its integrity has been maintained by verifying an internal hash digest within the plaintext. MULTOS confirms that these MSM controls were intended for this MCD by comparing the MCD id contained in the data with its own MCD id, and installs MSM Controls Data in its non-volatile memory if so. This ensures the correct MSM Controls Data is loaded onto the correct MCD. The presence of MSM Controls Data then allows authorised applications to be loaded onto the MCD.

3.8.1.3 Key Installation is regarded as a sensitive operation, so MULTOS strictly limits the number of unsuccessful attempts to perform the operation that can be made in the life of the MCD.

3.8.2 Specification

3.8.2.1 The MCD shall be able to load protected MSM Controls Data.

3.8.2.2 The MCD shall only load its own unique MSM Controls Data.

3.8.2.3 The MCD shall only allow MSM Controls Data to be loaded once.

3.8.2.4 Key Installation is a sensitive function. MULTOS shall keep a count of the number of failed attempts to execute this function. If this count reaches a defined limit, MULTOS shall permanently disable this function.

3.8.3 Description and Explanation

The MCD shall be able to load protected MSM Controls Data.

3.8.3.1 During implementation of MULTOS in silicon for the target processor, MULTOS security data is injected into non-volatile memory. The MULTOS security data includes a MCD-unique identifier (the MCD id) and MCD-unique symmetric transport key (tkv).

3.8.3.2 MSM Controls Data for a specific MCD includes the MCD-unique identifier, the MCD's permissions (defined in section 3.2) and the MCD-unique transport key (mkd). MSM Controls Data is encrypted by the MSM using the MCD-unique symmetric transport key (tkv). MSM Controls Data is provided to the MCD Issuer for loading on the target MCD.

3.8.3.3 The MCD Issuer presents the MSM Controls Data to the target MCD. This is done by submitting the Set MSM Controls command to MULTOS via an IFD. MULTOS checks a flag in the MULTOS security data to ensure that MSM Controls Data has not already been loaded successfully. MULTOS then decrypts the MSM Controls Data using tkv, which is stored in non-volatile memory. If the decrypted MCD-unique identifier does not match the identifier stored in non-volatile memory, the MSM Controls Data is rejected.

3.8.3.4 MSM Controls Data is required to support the Application Load and Authentication SEF and Application Transport Confidentiality SEF.

The MCD shall only load its own unique MSM Controls Data.

3.8.3.5 Since MSM Controls Data is encrypted using a symmetric key specific to the target MCD, only the target MCD is able to decrypt the data and load it successfully. Furthermore, the MCD is able to load only its own MSM Controls Data, since it will not be able to decrypt any other MCD's MSM Controls Data.

3.8.3.6 This ensures a MCD cannot load MSM Controls Data intended for another MCD and therefore cannot be made to masquerade as another MCD (e.g., in order to load applications not intended for it).

The MCD shall only allow MSM Controls Data to be loaded once.

3.8.3.7 If the MSM Controls Data is loaded successfully, MULTOS sets the flag in MULTOS security data to indicate this has occurred.

3.8.3.8 This ensures that once the MCD Issuer has enabled and issued the MCD, bogus MSM Controls Data cannot be created and loaded onto the MCD.

Key Installation is a sensitive function. MULTOS shall keep a count of the number of failed attempts to execute sensitive functions. If this count reaches a defined limit, MULTOS shall permanently disable the MCD.

3.8.3.9 MULTOS treats Application Load and Authentication, Application Transport Confidentiality, Application Deletion and Key Installation as sensitive functions. MULTOS maintains a count of the number of failed attempts to execute a sensitive function that have been made in the life of the MCD. If the total reaches six (across all sensitive functions, not six attempts per sensitive function), MULTOS permanently disables the MCD (i.e., MULTOS will no longer respond to any commands or other inputs received from an IFD). This is done in order to prevent brute force or exhaustion attacks on sensitive functions.

3.8.4 Security Mechanisms

The Key Installation SEF shall use the following security mechanism:

a) symmetric DES decryption, to recover the contents of MSM Controls Data.

3.8.5 Threats Addressed

3.8.5.1 The Key Installation SEF supports the Application Load and Authentication SEF in addressing the following threat:

- T1: Unauthorised load of an application onto a smart card.

3.8.5.2 In order to address threat T1, the Application Load and Authentication SEF relies, in part, on the MCD's permissions being securely stored on the MCD. The Key Installation SEF provides the function to securely store the MCD's permissions on the MCD.

3.8.5.3 The Key Installation SEF supports the Application Transport Confidentiality SEF in addressing the following threats:

- T4: Application is loaded onto an unauthorised smart card
- T7: Unauthorised access is gained to sensitive information held within a smart card application.

3.8.5.4 In order to address Threats T4 and T7, the Application Transport Confidentiality SEF relies on the MCD-specific secret transport key (mkd_sk) being securely stored on the MCD. The Key Installation SEF provides the function to securely store mkd_sk on the MCD.

3.9 Cryptography Control SEF

3.9.1 Overview

3.9.1.1 The Cryptography Control SEF correlates to Method of Use U5 and supports the Auth_Obj_1 security objective.

3.9.1.2 MULTOS allows certain features provided by the MCD to be regulated so that only authorised applications can access them. In the version of MULTOS subject to evaluation, the regulated features are strong cryptography mechanisms provided by the MCD. An application must be authorised by the MSM in order to be able to use strong cryptography (this may be necessary, for example, where government controls determine the right to access strong cryptography). An application's permission to access regulated features is included in the ALC. Also recorded in the ALC is a one-way cryptographic hash of the application's code.

3.9.1.3 MULTOS records the application's permission to use a regulated feature with the application when the application is loaded. MULTOS also calculates a hash value of the code segment of the application and compares it with the hash value in the ALC, to ensure the application's code has not been altered. Every attempt by the application to use a regulated feature is checked by MULTOS and permission is granted or denied accordingly. In this way, MULTOS controls an application's access to regulated features of the MCD.

3.9.2 Specification

3.9.2.1 MULTOS shall ensure only applications specifically authorised by the MSM can access strong cryptography primitives.

3.9.2.2 MULTOS shall support a means to ensure an application's code loaded onto the MCD is the same as the code originally approved for access to strong cryptography primitives.

3.9.3 Description and Explanation

MULTOS shall ensure only applications specifically authorised by the MSM can access strong cryptography primitives.

3.9.3.1 The ALC contains a flag indicating whether or not the application is authorised to use MULTOS's strong cryptography primitives. This information is stored with the application when it is loaded onto the MCD. Every time an application attempts to call a strong cryptography primitive, MULTOS checks the control flag to determine if the application is allowed to make the call. If it is, MULTOS will process the call. If the flag indicates access is not authorised, MULTOS will return an error condition to the application.

3.9.3.2 The MSM wishes to control which applications can access strong cryptography. This is necessary to comply with government restrictions on the use by Application Writers of strong cryptography. An Application Writer must obtain appropriate documentation (e.g., an export licence) from the appropriate government body before the MSM will authorise the application's use of strong cryptography. The MSM authorises an application to use strong cryptography by digitally signing its ALC with the cryptography access flag set to allowed.

MULTOS shall support a means to ensure an application's code loaded onto the MCD is the same as the code originally approved for access to strong cryptography primitives.

3.9.3.3 If an application is approved to use strong cryptography, a one-way cryptographic hash of its code segment is created and submitted to the MSM as part of the application details when the MCD Issuer requests ALCs. MSM authorises the hash value by including it in the signed ALC. When the application is loaded on to a MCD, MULTOS calculates the hash value over the application's code segment and compares it with the hash value contained in the ALC. If the two values differ, the load request is rejected. Although this mechanism is provided to support the Cryptography Control SEF, it is implemented as part of the Application Load and Authentication SEF (see section 3.2).

3.9.3.4 Although the Application Writer is responsible for obtaining approval to use strong cryptography in an application, the MCD Issuer determines if application authentication will be applied. Therefore, if the MCD Issuer wished to change the application's code, it could do so, specify that application authentication was not required and then load arrange for an application not approved for strong cryptography to be loaded onto a MCD. The hash value mechanism is designed to counter this specific realisation of Threat T8. This mechanism depends on appropriate supporting procedures, to ensure MSM is provided with a trustworthy hash value for inclusion in the ALC.

3.9.4 Security Mechanisms

The Cryptography Control SEF shall use the following security mechanisms:

a) one-way cryptographic hash of application's code segment, to verify the application loaded on the MCD is the same as the application approved for strong cryptography

b) a flag indicating whether or not an application is approved to use strong cryptography, which is checked whenever an application attempts to call a strong cryptography primitive.

3.9.5 Threats Addressed

3.9.5.1 The Cryptography Control SEF addresses the following threat:

- T8: Unauthorised use of strong cryptography.

3.9.5.2 This threat is addressed by the following mechanisms:

a) defininition of a flag in the ALC which indicates whether or not an application can access strong cryptography

b) inclusion in the ALC of a trustworthy one-way crytographic hash of the approved application's code segment.

3.9.5.3 The application can only be loaded onto a MCD if its ALC has been digitally signed by the MSM. By signing the ALC, the MSM authorises the application to use strong cryptography and authorises its code segment hash value. MULTOS:

a) confirms the application has been authorised by the MSM before loading the application onto the MCD

b) confirms the validity of the hash value by calculating its own hash value over the loaded application's code segment

c) mediates all attempts by the successfully loaded application to access strong cryptography.