ID-One Cosmo 32 v5

FIPS 140-2 Level 3

Security Policy

Public Version

Revision R01

October 27, 2005

Oberthur Card Systems
4250 Pleasant Valley Road
Chantilly, VA 20151-1221
(703) 263 0100

Version Control

Table 1 shows the version history of this Security Policy.

Version - Date	Description
R01 October 27 2005	Official Release of the public version for FIPS validation

Table 1 - Document Version History

TABLE OF CONTENTS

1 Introduction

This document defines the Security Policy for the Oberthur Card Systems ID-One Cosmo 32 v5 Cryptographic Module, hereafter referred to as the module. In the scope of this document, the cryptographic module is a single chip Integrated Circuit with its embedded firmware. It is designed to be encased in a hard opaque resin which can be embedded into a plastic card, or any other identity token, to produce the ID-One Cosmo 32 v5 JavaCard Chip Platform, on which FIPS 140-2 Level 3 validated applets may be loaded and instantiated at post issuance.

The cryptographic module is submitted for validation in accordance with FIPS 140-2 Level 3 standard.

Included are a description of the security requirements of the module and a qualitative description of how each security requirements is achieved. In particular, this security policy specifies the security rules under which the cryptographic module must operate.

2 Applicable Documents

- Open Platform Card Specification - Version 2.1.1 – Mars 2003, Global Platform
- Open Platform Card Specification - Version 2.1.1 Amendment A – February 2004, Global Platform
- Java Card TM 2.2 Virtual Machine Specification – June 2002, Sun Microsystems
- Java Card TM 2.2 Application Programming Interface – revision 1.1- September 2002, Sun Microsystems
- Java Card TM 2.2 Runtime Environment Specification - June 2002, Sun Microsystems
- Global Platform 2.1 Card Implementation Requirements –May 2002, Visa International
- Visa Open Platform Card Implementation Requirements Configuration 3 – Multiple Security Domains with DAP Capability October 2001
- Visa Open Platform Card Implementation Requirements Configuration 3 – Multiple Security Domains with DAP Capability Version 2 – Errata February 2002
- [FIPS140-2] National Institute of Standards and Technology, FIPS 140 -2 standard.
- [FIPS140-2A] National Institute of Standards and Technology, FIPS 140 -2 Annex A: Approved Security Functions.
- [FIPS140-2B] National Institute of Standards and Technology, FIPS 140 -2 Annex B: Approved Protection Profiles,
- [FIPS140-2C] National Institute of Standards and Technology, FIPS 140 -2 Annex C: Approved Random Number Generators
- [FIPS140-2D] National Institute of Standards and Technology, FIPS 140 -2 Annex D: Approved Key Establishment Techniques
- [DES] National Institute of Standards and Technology, Data Encryption Standard, Federal Information Processing Standards Publication 46-3, October 25, 1999.

- [DES Modes] National Institute of Standards and Technology, DES Modes of Operation, Federal Information Processing Standards Publication 81, December 2, 1980.
- JC2.2 API SRS revision issuee1-AC, Oberthur Card Systems
- Basic Input/Output System (BIOS) SRS, revision 1-AA, Oberthur Card Systems
- JavaCard Virtual Machine V2.2 SRS, revision 1-AB, Oberthur Card Systems
- "Integrated circuit(s) cards with contacts - Part 2 Dimension and Location of the contacts." ISO/IEC 7816-2 (1999)
- "Integrated circuit(s) cards with contacts - Part 3 Electronic signal and transmission protocols." ISO/IEC 7816-3 (1997), ISO/IEC 7816-3 AMD1 (2002)
- "Integrated circuit(s) cards with contacts - Part 4: Inter industry commands for interchange." ISO/IEC 7816-4 (1995), ISO/IEC 7816-4 AMD1 (1997)
- "Numbering system and registration procedure for application identifiers" ISO/IEC 7816-5 (1994), ISO/IEC 7816-5 AMD1 (1996)
- “Information technology – Security techniques – Digital signature scheme giving message recovery - Part 2: Mechanism using a hash function.” ISO/IEC 9796-2 (1997)
- “Information technology – Security techniques – Message Authentication Codes – Part 1: Mechanisms using a block cipher” ISO/IEC 9797-1 (1999)
- Contactless integrated circuit(s) cards – Proximity cards — Part 2: Part 2: Radio frequency power and signal interface, ISO/IEC 14443-2 (2001)
- Contactless integrated circuit(s) cards – Proximity cards — Part 3: Initialization and anti-collision, ISO/IEC 14443-3 (2001)
- Contactless integrated circuit(s) cards – Proximity cards — Part 4: Part 4: Transmission protocol, ISO/IEC 14443-4 (2001)
- "Integrated Circuit Card Specifications for Payment Systems" – EMV 2000
    - Part 1: Electromechanical Characteristics, Logical Interface, and Transmission Protocols (version 3.0)
    - Part 2: Data Elements and Commands (version 3.0)
    - Part 3: Application Selection (version 3.0)
    - Part 4: Security Aspects (Version 3.0)
- "API File System Library" Ref: 055731 00 SRS revision-issue 1-AA, Oberthur Card Systems
- "API Utils File System" Ref: 055901 00 SRS revision-issue 1-AA, Oberthur Card Systems
- "Java Card 2.2 Biometry API proposal" Javadoc version (4-4-02) on JCF web site
- "Format des templates biométriques" FQR 110 1767 Ed 1, Oberthur Card Systems

3 Definitions and Acronyms

3.1 Definitions

3.1.1 Card Manager

The Card Manager, also called Issuer Security Domain, is the on-card representative of the Card Issuer (Cryptographic Officer). It is the most privileged entity of the cryptographic module as it is the only entity that performs Card Content management without having been explicitly delegated previously. Privileges of the Card Manager include but are not limited to card locking, card termination, CVM (Card Holder Verification Method) management, and multiple selections (through logical channels).

The Issuer Security Domain shall have the following set of privileges clearly identifying its functionality (i.e. a Security Domain with card lock, card terminate and CVM management privileges and possibly the Default Selected privilege) in addition to its implied unrestricted Card Content management privilege. If the card supports Supplementary Logical Channels, the Issuer Security Domain shall also have the multiple selection privilege.

3.1.2 Security Domains

Security Domains allow a number of distinct identities to be established on the ID-One Cosmo 32 v5 platform. These are identities that control access to the various applets stored on the module. A Security Domain represents the identity of an application (applet) operator.

3.1.3 Applets

“Applets” are applications that can be executed on the Chip platform. They come in two parts; the applet executable code, which defines all the functions that could be executed on the chip platform, and the Applet Instance, that provides the environment (i.e. variables) and an interface to the functions present in the applet executable code. An applet can have several instances, each with its own variables, but all sharing the same functionality as defined in the underlying executable code. The Applet Instance is the mandatory communication path between the applet Executable Module and the outside world.

In order for an application to be activated and provide its high level services to the outside world, two prerequisites must have been fulfilled:

1. The Applet Executable Load File, that contains the actual Java code (Executable Module) of the application, must be present on the chip platform. This can be achieved by physically downloading the load file into the chip platform EEPROM, or by activating a pre-loaded Executable Load File present in ROM.
2. At least one applet instance of the executable module must have been created.

The services described in this Security Policy allow the security officer to load and unload (delete) any applets. This allows the loading of executable load files, which can take up to 30 seconds depending of the size of the file, to take place during pre-issuance. Until the time they are instantiated, the executable load files can be considered as “dead code”. The actual applet activation, which is done through instantiation, takes only a few milliseconds and could take place in post issuance, under the control of the Security Officer, and after the applet has been FIPS 140-2 validated.

For the cryptographic module to be correctly operated according to this Security Policy, applets instantiated into the chip platform must be validated to FIPS 140-2.

3.2 Acronyms

AES	Advanced Encryption Standard
AID	Application Identifier
AP	Application Provider
APDU	Application Protocol Data Unit
API	Application Programming Interface
ATR	Answer To Reset (contact mode)
ATS	Answer to Select (contactless mode)
API	Application Programming Interface
CBC	Cipher Block Chaining
CRC	Cyclic Redundancy Check
DAP	Data Authentication Pattern
DES	Data Encryption Standard
DPA	Differential Power Analysis
DM	Delegated Management
DRNG	Deterministic Random Number Generator
ECB	Electronic Code Book
EEPROM	Electrically Erasable and Programmable Read Only Memory
EMI	Electromagnetic Interference
EMC	Electromagnetic Compatibility
ICAO	International Civil Aviation Organization
ISO	International Standard Organization
JC	Java Card ™
JCRE	Java Card ™ Runtime Environment
MAC	Message Authentication Code
NDRNG	Non Deterministic Random Number Generator
OP	Open Platform
PIN	Personal Identification Number
PKCS	Public Key Cryptographic Standards
RAM	Random Access Memory
ROM	Read only Memory
RSA	Public key cryptographic algorithm invented by Rivest, Shamir and Adleman
SHA	Secure Hash Algorithm
SPA	Simple Power Analysis
TDES	Triple DES
TLV	Tag Length Value

4 Module Overview

4.1 Product overview

The Oberthur Card Systems ID-One Cosmo 32 v5 is a single chip multi-application cryptographic JavaCard module specifically designed for identity and government market needs.

The cryptographic module loads and runs applets written in Java programming language. It includes a native implementation of the latest Java Card™ (version 2.2) and Open Platform (Version 2.1.1A) specifications, with full support for Delegated Management and DAP / Mandated DAP, that define a secure infrastructure for post-issuance programmable platforms.

Additional features include biometric extensions as defined by the JavaCard Forum and an on card fingerprint matching using matching algorithms from various third parties.

The built in management of Logical Channels allows the module to support multiple applications simultaneously, each with their own Security Domain.

The ID-One Cosmo 32 v5 provides tamper-resistance and tamper evidence features that meet FIPS 140-2 Level 3 physical requirement.

The module requires a lower voltage than traditional smart cards to operate making it the perfect cryptographic module for a new range of application using lower voltage portable readers. The cryptographic module operates under either 5 Volt power supply (ISO 7816-3 Class A) or 3 Volt power supply (ISO 7816-3 Class B).

4.2 Cryptographic Algorithms

The following algorithms are available on the ID-One Cosmo 32 v5 cryptographic module:

- DES encryption and decryption (ECB & CBC modes)[1]
- DES Message Authentication Code generation and verification[2]
- Triple DES encryption and decryption (ECB & CBC modes) using 128-bit and 192-bit key sizes:
    - In Raw mode (no formatting)
    - With ISO/IEC 9797 automatic padding, methods 1.
    - With ISO/IEC 9797 automatic padding, methods 2.
- Triple DES Message Authentication Code generation and verification:
    - In Raw mode (no formatting)
    - With automatic Padding from ISO/IEC 9797 methods 1,
    - With automatic Padding from ISO/IEC 9797 methods 2,
    - With automatic Padding from ISO/IEC 9797 methods 2 and with MAC algorithm 3,

[1] For legacy systems only. Transitional phase only valid until May 19, 2007.
[2] For legacy systems only. Transitional phase only valid until May 19, 2007.

- RSA key generation (1024 to 2048-bit key size by 64 bit increments)
- RSA encryption and decryption (key Wrapping/Unwrapping)[3]:
  - In Raw mode (no formatting)
  - With PKCS#1 automatic padding,
  - With PKCS#1-OAEP automatic padding,
- RSA signature and verification:
  - In Raw mode (no formatting)
  - With PKCS#1 automatic padding,
  - With PKCS#1-PSS automatic padding,
  - With ISO/IEC 9796 automatic padding,
- SHA-1 digest computation
- MD5[4] digest computation.
- RNG

[3] In this FIPS configuration, without an application (applet) loaded, the only available service which utilizes the RSA algorithm is one that uses a 1024 bit RSA public key for DAP verification (see section 10.2.3, DAP VERIFICATION). As a result, the use of RSA for encryption, decryption, key wrapping and unwrapping, is not available in this module.

[4] MD5 is not available as a service in this module.

4.3 Security Policy Overview

This document defines the Security Policy for the Oberthur Card Systems ID-One Cosmo 32 v5 Cryptographic Module. In the scope of this document, the cryptographic module is a single chip Integrated Circuit with its embedded firmware. It is designed to be encased in a hard opaque resin which can be embedded into a plastic card, or any other support to produce the ID-One Cosmo 32 v5 JavaCard Chip Platform, on which FIPS 140-2 Level 3 validated applets may be loaded and instantiated at post issuance.

The photo, Figure # 1, shows an example of the target of evaluation.

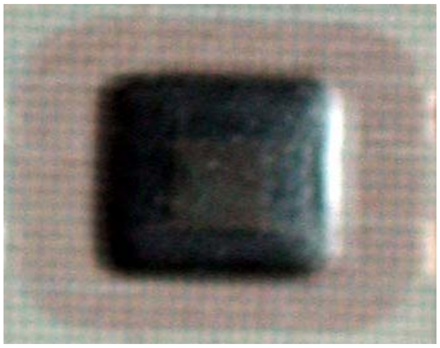

Figure 1

The diagram, Figure # 2, shows an example the integrated Circuit (micro-controller) and the golden wires underneath the epoxy resin.

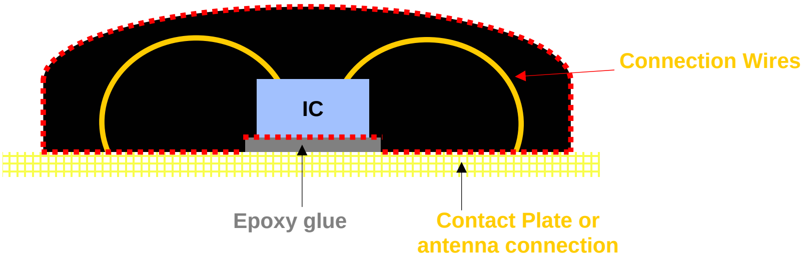

Figure 2

The red dotted line shows the module cryptographic boundary. The epoxy glue and the support on which the crypto module is glued (contact plate or antenna) are not part of the crypto module boundary.

The module contains a microprocessor and EEPROM to provide processing capabilities and data storage and offers Java Card™ Technology and Open Platform services to applets on the chip.

This document addresses the submission for validation of the module in accordance with FIPS 140-2 Level 3 standard.

Included are a description of the basic security requirements for the module and a qualitative description of how each security requirement is achieved. In particular, this security policy specifies the security rules under which the cryptographic module must operate.

The Chip platform directly provides all the low-level services such as memory management, I/O control, cryptographic algorithms and physical security. It also contains a native implementation of the Java Card ™ specification (JC) version 2.2 and of the Open Platform (OP) version 2.1.1A specification, which define a secure infrastructure for post-issuance programmable platforms. These services can be accessed by the applets instantiated from code loaded onto the chip EEPROM or ROM using the Java Card™ Application Programming Interface (API).

The Card Manager provides the Open Platform services that are both internal (accessible by applets) and external services (accessible by external or non-chip applications).

This validation is aimed at the Systems software, virtual machine, and Card Manager/Security Domains applets without any other instantiated applets.

Applets subsequently instantiated, would have to be FIPS 140-2 Level 3 validated for the resulting ID-One Cosmo 32 v5 chip platform with applets to remain FIPS 140-2 Level 3 validated.

If an applet, which is not FIPS validated with the same security level (i.e. Level 3), is instantiated on this module, the module loses its FIPS validation.

5 Module Versions

The ID-One Cosmo 32 v5 Module functionality can be extended through the use of a firmware extension called optional code. Such optional code can be loaded in the EEPROM only during manufacturing and cannot be subsequently removed or modified. Examples of functionality that can be added through such firmware extension include biometric match on card algorithms, support for elliptic curve cryptography, etc.

This document addresses the submission for validation of the module “ID-One Cosmo 32 v5” based on the following commercial configuration:

ID-One Cosmo 32 v5.3 FIPS: Hardware Platform # ‘90’ with Firmware # ‘E311-063842’.

5.1 Module Identification

Module identification and configuration (complete firmware version, including optional code extension as well as the status of some locks set during module pre-issuance that allow to activate FIPS mode) can be retrieved at any time using the Get Data services described in paragraph 10.2.4. The associated tag is ‘DF52’ and the return value to check for FIPS mode is:

‘DF52 xx 00 02 **pppp** 01 06 xxxxxxxxxxxx 02 01 **qq** 03 06 **ssssss**xxxxxx’

Where:

- **pppp** is the primary Firmware version (E311 for ID-One Cosmo 32 v5)
- **qq** is the status of the FIPS locks. Should be ‘**00’** for the module to be in FIPS mode.
- **ssssss** is the Firmware extension number coded in BCD (‘063842’ in this FIPS configuration). On the FIPS 140-2 certificate from NIST, this number is separated from the primary firmware version with a dash (e;g. ‘E311-063842’)

6 Security Level

The Oberthur Card Systems ID-One Cosmo 32 v5 cryptographic module meets the overall requirements applicable to Level 3 security of FIPS 140-2.

Security Requirements Section	Level
Cryptographic Module Specification	3
Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	NA
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

Table 2 - Module Security Level Specification

7 Modes of Operation

The ID-One Cosmo 32 v5 described in this security policy does not include any non-FIPS validated applet instances. As such, the cryptographic module is always in an approved mode of operation. The security services described in this document can be used to load any kind of applets into the ID-One Cosmo 32 v5 JavaCard chip platform. However, it is the responsibility of the Cryptographic Officer to insure that only FIPS validated instances are created. See 3.1.3 Applets.

The FIPS approved mode of operation for the evaluation described in this Security Policy starts from the instantiation of the Card Manager and Security Domains and ends with the instantiation of any non-FIPS validated applets or the instantiation of a FIPS validated applet that has its Access Control Rules (ACRs) set improperly.

The Cryptographic Officer can determine whether the card is still in FIPS mode by authenticating to the Card Manager and issuing a Get Status command to list all the applications instances currently installed in the card. However, with Java card, an application can be given any AID (application Identifier) during instantiation, regardless of the identity of the underneath executable load file. To prevent a non FIPS approved applet to be instantiated and given the AID of an approved applet, Oberthur has implemented a special Get Status command that returns not only the AID given to the applet instance, but also, and more important, the AID and version number of the underlying executable load file. The Cryptographic Officer can then at any time check that only FIPS approved executable load files have been instantiated.

The Get Data command on a selected applet instance, allows the Cryptographic Officer to verify that the ACRs have been set correctly.

8 Cryptographic Module Specification

The ID-One Cosmo 32 v5 cryptographic module supports a command set aimed at allowing the mutual authentication of identities using strong cryptography with “card acceptance devices” in ISO mode (and PCs or other terminals that they might be connected to). Specifically, the TDES algorithm is used within authentication commands between the cryptographic module and the “card acceptance device” environment for strong authentication of identities. Establishment of identities using these commands is then used to fulfill “access conditions” which limit the ability of the external world to access information and/or commands on the module.

ID-One Cosmo 32 v5 adheres to ISO/IEC 7816 specifications for Integrated Circuit Chip (ICC) based identification cards and to ISO/IEC 14443 for contactless operations. The “cryptographic boundary” for the ID-One Cosmo 32 v5 module vis-à-vis the FIPS 140-2 validation is the “module edge”. The module is the encapsulated chip and is constructed to provide tamper resistance and tamper evidence required in the FIPS 140-2 physical Level 3 validation.

This validation is aimed at the Systems software, virtual machine, and Card Manager/Security Domains applets without any other instantiated applets. Applets subsequently instantiated, would have to be FIPS 140-2 Level 3 validated for the resulting ID-One Cosmo 32 v5 chip platform with applets to remain FIPS 140-2 Level 3 validated.

The Oberthur Card Systems ID-One Cosmo 32 v5 is a single chip implementation of a cryptographic module. The module comprises the following elements:

- Secure micro controller Integrated Circuit with:
  - A 32 bit crypto coprocessor optimized for public key cryptographic calculations
  - A Triple DES (Data Encryption Standard) Co-Processor
  - User ROM and RAM
  - High reliability 36 KB EEPROM for both customer applications and Operating System data
  - System firmware, consisting of the operating system installed in Read Only Memory (ROM)
  - Optional Code as identified in section 5 Module Versions
- Applets (Applications) that are to be installed onto the module.
- Critical Security Parameters stored in EEPROM as part of the chip platform personalization operation.

Figure 3 demonstrates a logical block diagram of the module.

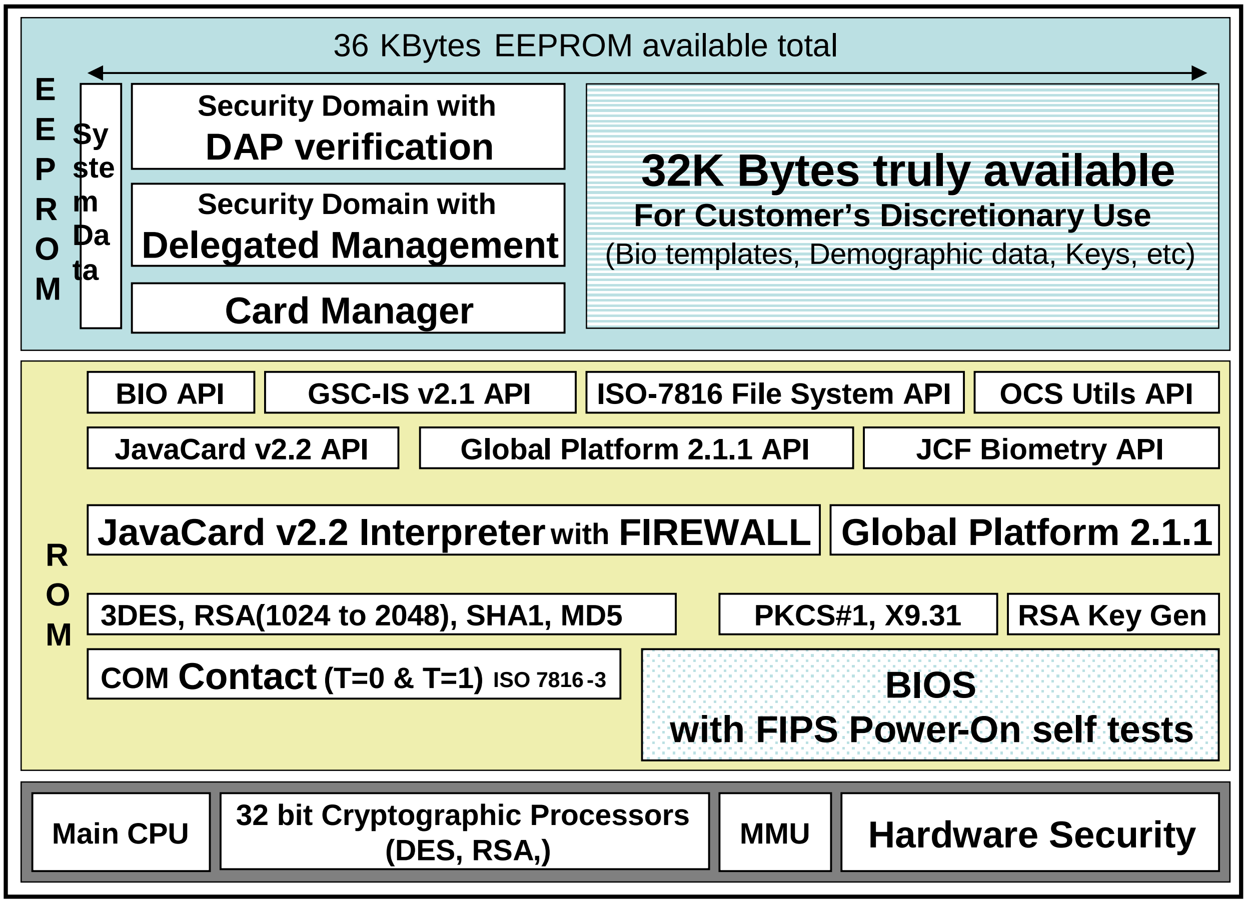
36 KBytes EEPROM available total
EEPROM
System Data
Security Domain with DAP verification
Security Domain with Delegated Management
Card Manager
32K Bytes truly available
For Customer's Discretionary Use
(Bio templates, Demographic data, Keys, etc)
ROM
BIO API
GSC-IS v2.1 API
ISO-7816 File System API
OCS Utils API
JavaCard v2.2 API
Global Platform 2.1.1 API
JCF Biometry API
JavaCard v2.2 Interpreter with FIREWALL
Global Platform 2.1.1
3DES, RSA(1024 to 2048), SHA1, MD5
PKCS#1, X9.31
RSA Key Gen
COM Contact (T=0 & T=1) ISO 7816-3
BIOS
with FIPS Power-On self tests
Main CPU
32 bit Cryptographic Processors (DES, RSA,)
MMU
Hardware Security

Figure 3: Logical Block Diagram of the Oberthur ID-One Cosmo 32 v5

9 Ports and Interfaces

The integrated circuit used in the ID-One Cosmo 32 v5 can support the following interfaces:

- ISO/IEC 7816: Identification Cards - Integrated Circuit Cards with Contacts

9.1 ISO/IEC 7816 Physical Interface (contact mode)

9.1.1 Interface Physical Specifications

In this contact mode, communication to and from the cryptographic module is done through a printed circuit (contact plate) that provides the electrical connection required. Five electric wires connect the module to the printed circuit, and from there, to the outside world. The printed circuit itself is outside of the module cryptographic boundaries and mentioned only for illustration purposes.

The ID-One Cosmo 32 v5 operates in both ISO 7816-3 class A and class B. Class A requires a power supply voltage between 4.5 Volt and 5.5 Volt. Class B requires a power supply voltage between 2.7 Volt and 3.3 Volt. This opens new ranges of application using lower voltage portable readers.

9.1.2 Interface Electrical Specifications

The following picture shows an example of printed circuit and the location were the five electrical connections from the module are wire bonded to the contact plate.

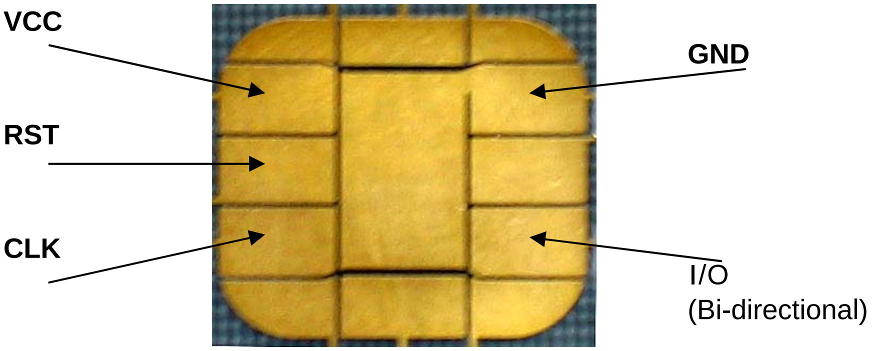

Figure 4: Example of contact plate used to provide electrical communication with the cryptographic module

Contacts that are not identified do not carry any signals and are either grounded or left not connected.

The 5 electrical signals transmitted to the module through the contact mode wires coming from the contact plate are the following:

- **VCC**: Supply Voltage Power supply input. (1.62V to 5.5V)
- **GND**: Ground (reference voltage)

- **RST**: External reset signal from the interface device (card read / write device)
- **CLK**: External clock (1MHz to 10MHz). This clock is just for data transmission as both processor and coprocessors are driven independently by an internal oscillator at a much higher frequency.
- **I/O**: Input or output for serial data to / from the processor

These 5 electronic signals are in full compliance with ISO/IEC 7816-3 standard.

9.1.3 Transmission protocol and speed

The transmission protocols with the ID-One Cosmo 32 v5 comply with ISO/IEC 7816-3 (half duplex character oriented transmission protocols ISO T=0 and T=1).

Characters can be exchanged in direct convention (Z level corresponds to a logical 1 and LSB is sent first) or in inverse convention (Z level corresponds to a logical 0 and LSB is sent first).

The Oberthur ID-One Cosmo 32 v5 supports the Protocol and Parameter Selection to select a new protocol type or change transmission baud rate.

Up to 256 data bytes can be exchanged within one command.

The maximum communication speed in contact mode is 614,400 bauds (with a clock of 4.9Mhz).

9.2 Logical Interface Description

Once communication is established between the reader and the platform, the platform functions as a "slave" processor to implement and respond to the reader commands. The platform adheres to a well-defined set of state transitions. Within each state, a specific set of commands is accessible. The I/O port of the platform provides the following logical interfaces:

	ISO 7816 (Contact mode)
Data In:	I/O Pin
Data Out:	I/O Pin
Status Out:	I/O Pin
Control In:	I/O, Clock and Reset Pins

Synchronization timing controls provided in part by the platform CLK clock input, manage the separation of these logical interfaces that use the same physical port.

10 Roles & Services

10.1 Roles

The Oberthur ID-One Cosmo 32 v5 defines two distinct roles that are supported by the internal cryptographic system: the Cryptographic Officer and the User.

- ***Cryptographic Officer***: This role is responsible for administrating the cryptographic platform and managing its overall security configuration. The Cryptographic Officer establishes his identity by establishing a secure channel with the Card Manager Security Domain, thus providing knowledge of the Card Manager key set. The role of Cryptographic Officer is generally assumed by the Card[5] Issuer.

- ***User/Applet provider***: The Applet Provider is the applet developer that uses the Java API, available on the module. The developer is regarded as an internal user to the platform. The cryptographic services provided by the module are delivered through the use of well-documented APIs. An applet can have a dedicated security domain instance (Applet Provider Security Domain), or may rely on the Card Manager Security Domain.

10.1.1 Identity based Authentication

- ***Identification****.* The operator identifies him/herself by selecting the application and a key set associated with the application. The application of the Cryptographic Officer is the Card manager. The application of the applet providers is their own applet. The selection of the application is done by a SELECT command. The selection of the key set is done through the INITIALIZE UPDATE command. (The same command that will be used to start the authentication that follows the identification.)

- ***Authentication****.* The operator authenticates him/herself using a mutual authentication comprising two commands INITIALIZE UPDATE and EXTERNAL AUTHENTICATION. During this mutual authentication, the operator has to encrypt a message sent by the card, proving knowledge of the TDES key set that was referenced during the identification.

Each INITIALIZE UPDATE must be immediately followed by a successful EXTERNAL AUTHENTICATE command. Otherwise, the event is recorded in the card Audit Log and the next Initialize Update performed on the same key set will be exponentially slowed down to discourage attacks. This provides a strong protection against brute force attacks as no more than a few consecutive unsuccessful authentication attempts are possible within one minute.

The authentication remains valid until the next Identification phase (SELECT command) or until an unsuccessful authentication or a card reset (power-off) has been initiated.

10.1.2 Logical Channels

The Oberthur ID-One Cosmo 32 v5 module provides a full support for Logical Secure Channels as defined by GP2.1.1. Secure channel protocol is used to establish a secure communication channel between the module and an external entity during an Application Session.

[5] In this security policy, the term "Card Issuer" refers to the entity who issue the cryptographic module to the end user regardless of any form factor (card, ePassport, token, etc…)

Logical Channels facilitate the possibility of more than one of the above external entities to communicate concurrently with multiple applications on the card, each within its own logical secure environment.

10.2 Services

10.2.1 Cryptographic Officer Services

Several services are made available to an authenticated Cryptographic Officer only. They are primarily used to manage Security Domains and allow the loading of applets into the card.

INSTALL: Installing an application or Security Domain requires the invocation of several different on-card functions (e.g. the install method). The INSTALL command is used to instruct the Card Manager or Security Domain with the Delegated management privilege as to which installation step it shall perform during an application installation process. The command must be used in the context of a Secure Channel and so its level of security must match the security level defined in the EXTERNAL AUTHENTICATE command.

LOAD*:* This command is used to load the byte-codes of the Load File (package) defined in the previously issued INSTALL command. Applets are loaded inside a Secure Channel established by the Crypto-Officer with the Card Manager during the identification/authentication process. The applet is divided into a series of blocks that fit in a LOAD command. The loading consists of a series of LOAD commands, each one transmitting a block, encrypted and followed by a TDES MAC with the TDES key set selected by the Crypto-Officer during the identification process. The TDES MAC ensures the correct transmission of each block of the applet, therefore ensuring the correct transmission of the whole applet.

DELETE: This command is used to delete a uniquely identifiable object. The object may be an Application, a load file, or a Key-Set. The command must be used in the context of a *Secure Channel* and so its level of security must match the security level defined in the EXTERNAL AUTHENTICATE command. The deletion of an object includes the complete zeroization of the memory zone RAM and/or EEPROM) where it was stored. This is achieved by the Oberthur Garbage Collector that writes binary zeros in the associated memory zones and FAT (File Allocation Table) entry. The Garbage Collector is automatically called upon completion of the delete command.

PIN UNBLOCK: The Pin Unblock instruction is used to unblock the current global PIN. The command is used with Secure Messaging in the context of a Secure Channel; its level of security must so match the security level of the current Secure Channel.

PUT PUBLIC KEY: This command is used to load RSA Public keys such as the Token Verification Key or the DAP Verification Key. These keys are used for Delegated management (see below).

DELEGATE MANAGEMENT: Delegated Management gives the Card Issuer (Crypto-Officer) the possibility of empowering partnered Application Providers (Users) the ability to initiate approved and pre-authorized Card Content changes (loading, installation, extradition[6] or deletion). Please refer to section 14.3.2 for more details on how the Delegated Management works.

GET STATUS: The Get Status command is used to verify that the Card Manager has enabled all the access control rules. It allows retrieving of Card Manager and Application related life cycle status information according to a given match/searching criteria. This command can also be used by the Cryptographic officer to verify that the module is still in FIPS Mode and that only FIPS approved applications are instantiated. It's the complementary command to Set Status. The command must be used in the context of a Secure Channel and so its level of security must match the security level defined in the External Authenticate command.

[6] . Application Extradition allows an Application that is already associated with a Security Domain to be extradited and associated with another Security Domain

10.2.2 Cryptographic Officer and User Services

The following services (commands) are made available to authenticated Crypto-Officer and User/Applet Provider:

INITIALIZE UPDATE: This command initiates a Secure Channel used by subsequent commands. It allows platform authentication by the Crypto-Officer or User, and computes session keys used for MAC computation and command encryption in issuance of subsequent commands. However, the Secure Channel is not considered open until completion of a successful EXTERNAL AUTHENTICATE command that must immediately follow the INITIALIZE UPDATE command.

EXTERNAL AUTHENTICATE: This command is used by the module to authenticate the host, to establish the Secure Channel, and to determine the level of security required for all subsequent commands within the Secure Channel. The EXTERNAL AUTHENTICATE command must be preceded by the INITIALIZE UPDATE command.

STORE DATA: This command is used to personalize a data object from a Security Domain or to transfer data to an Application. The Issuer Security Domain determines if the command is intended for itself or an Application depending on a previously received command. If a preceding command was an INSTALL [for personalize] command, the Store Data command is destined for the Application identified in the data field of the command INSTALL [for personalize]. Multiple Store Data commands transfer data to the Application by breaking the data into smaller components for transmission. In this case, Store Data command is numbered (starting at '00' and increments by one in a sequential manner). The Issuer Security Domain transfers the Multiple Store Data to the Application, as long the numbering of the command is correct and no final block is received or no command INSTALL [for personalize] is issued. The command must be used in the context of a Secure Channel and the level of security for the command is dependent on the security level defined in the External Authenticate command.

The command Store Data, when used with the tag DF51h, enables the information recorded into the audit Log File to be erased. The use of the tag DF51h with this command is authorized only when the card manager is not LOCKED nor TERMINATED. Prior to reset audit log Data, an External Authentication must be successfully performed.

SET STATUS: This command is used to modify the life cycle state of the card or the life cycle of designated application. Set Status is the complementary command to Get Status. The command must be used in the context of a Secure Channel and so its level of security must match the security level defined in the External Authenticate command.

PUT TDES KEY: this command is used to:

1. Set the Initialization Secret Key (ISK);
2. Set the Receipt verification Secret Key (RSK);
3. Add a new key-set version containing a complete set of keys (Set of 3 double length DES keys);
4. Replace multiple keys within an existing key-set version;
5. Replace multiple keys within an existing key-set version and change its version number (Key-set replacement).

If it already exists in the Issuer Security Domain, the current key-set or specific key is replaced. A key is uniquely identified by the combination of its key-set version and its key index. An application may have multiple key-set versions. Multiple keys may exist within a given key-set version. The command must be used in the context of a Secure Channel and so its level of security must match the security level defined in the EXTERNAL AUTHENTICATE command to add or replace Security Domain key sets.

10.2.3 User Services

DAP VERIFICATION: DAP verification allows an Application Provider (User) to own a Security Domain which can be requested to check application code integrity and authenticity before the application code is loaded by the Crypto-Officer or any entity other than the Application Provider itself. More details on how DAP verification works can be found in section 14.3.3 Dap Verification.

10.2.4 No Role

The following services are available without authentication

GET DATA: The GET DATA command is used to retrieve a single data object, such as the Card Identification data that can be used to determine if the module manufacturing configuration is in approved mode (see section 5 Module Versions). This command can be used in clear mode outside of a Secure Channel. The following data objects can be retrieved using the get data command:

Tag (coded in **P1**, **P2**)	**Length** in bytes (coded in **Le**)	**Meaning**
0042h	Length of IIN + 2	IIN (new designation of BIN)
0045h	0Ah	Card Issuer Data (8 bytes)
DF64h	05h	Free EEPROM size
DF51h	Variable	Audit log
DF52h	Variable	Card Identification Data
DF53h	Variable	Java Configuration Data
00EEh	Variable	Card Profile Unique Identifier
9F7Fh	2Dh	Card production life cycle (42 bytes)
00CFh	0Ch	User Key diversification data (10 bytes)
00E0h	Variable	Key Information Template
00C1h	05h	Sequence Counter of the default Key Version Number (Implicit Key version)
00C2h	05h	Confirmation Sequence Counter (Delegated management)
0066h	Variable	Card Recognition Data

Table: Data object readable with Get Data command

SELECT: This command is used to select an application (Card Manager, Security Domain or Applet), and does not require prior authentication. This command also allows the identification of the Cryptographic Officer when issued with the Card Manager AID, and the identification of the User/Application Provider when issued with a Security Domain AID.

10.2.5 Relationship between Roles and Services

Roles/Services	Crypto-Officer (Card Manager)	User/Applet Providers (Security Domain)	Unauthenticated (Any role)
SELECT	X	X	X
INITIALIZE UPDATE	X	X	
EXTERNAL AUTHENTICATE	X	X	
STORE DATA	X	X	
GET DATA	X	X	X
SET STATUS	X	X	
GET STATUS	X		
PUT TDES KEY	X	X	
PUT PUBLIC KEY	X		
INSTALL	X		
LOAD	X		
DELETE	X		
PIN UNBLOCK	X		
DELEGATE MANAGEMENT	X		
DAP VERIFICATION		X	

11 Finite State Machine

The Open Platform Card Manager manages the states of the Java Card platform and applets life cycle. The cryptographic platform has its Card Manager in OP-Secured phase when issued to a user.

The Finite State Machine diagrams applicable to the module are provided as a separate document.

12 Physical Security

The ID-One Cosmo 32 v5 is a single chip cryptographic module. It is designed to meet FIPS 140-2 Level 3 requirements for physical security.

The module is a production quality IC. It meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. It uses standard passivation techniques for the entire chip.

In addition to the passivation material, a hard, opaque epoxy, that is resistant to commonly available solvents, is used to encapsulate the module into an opaque support.

The chip is usually in possession of either a Cryptographic Officer or of the User.

In order to physically attack the module, an attacker will have to take possession of the module and use extraordinary means such as electronic probe or electronic microscope.

As the chip module is covered with a hard, tamper-evident resin, that resin must be removed to attempt any physical attack on the chip.

In this event, the absence of the chip is easily detected by its owner. Once the chip has been attacked through extraordinarily physical means, the attack leaves permanent evidence and is consequently detected by the owner.

In addition to the above passivation material, the following active features available in the module provide increased protection against physical attacks:

- Low / high supply voltage sensor
- Low / high clock frequency sensor
- Low / high temperature sensor
- Light sensor
- Single fault injection (SFI) attack detection
- Programmable “Card Disable” feature

13 Operational Environment

During the manufacturing process, any applet executable load file can be loaded into the ID-One Cosmo 32 v5 JavaCard platform, but those files remain dead code until activated through an instantiation process, and only trusted (FIPS validated) applets can be instantiated. See section 3.1.3 Applets.

After completion of the manufacturing process (including pre-issuance), when the chip has reached its normal Operating Life Cycle State (Card Manager in Secured State), it is the responsibility of the Cryptographic Officer to insure that only FIPS validated instances are created.

The FIPS 140-2 Area 6 Operational Environment requirements are therefore not applicable.

14 Cryptographic Key Management

Details on the key management scheme are provided in a separate document.

The cryptographic module handles various keys and PIN

- Global PIN
- Card Manager and Security Domain Keys

14.1 Global PIN

The Global PIN (Personal Identification Number) supported by the ID-One Cosmo 32 v5 can be a sequence from 6 to 254 digits, or a passphrase of 127 characters max (any characters that could be coded in hexa on one byte). It may be used through a standard GP 2.1.1 API to authenticate the future Cardholder to the module with a probability of false authentication of less than 1/1,000,000. By successfully entering a PIN sequence, a cardholder can prove knowledge of a shared secret (the PIN) and thereby authenticate to the module.

The Cryptographic Officer has the capability to unlock a cryptographic module that has been lock after reaching a predefined number of consecutive errors on PIN verification. However, PIN setting and verification are available only through API to be called by an applet. Until such applet gets FIPS 140-2 validated, the Global PIN feature cannot be used.

14.2 Cryptographic Keys

The ID-One Cosmo 32 v5 in FIPS mode (i.e. in Card Manager OP-Secured) includes the following keys that conform to Global Platform Specifications v2.1:

14.2.1 Initial Issuer Transport Key

1. **KDC**: Initial Issuer Key set: Set of three Triple DES Keys (called KDC_{ENC} KDC_{MAC} and KDC_{KEK}) of 16 bytes each. The first two, KDC_{ENC} and $KDC_{MAC,}$ are only used to generate Secure Channel session keys during the initiation of a Global Platform Secure Channel, and the last one, KDC_{KEK} is used as a key transport key within a secure channel.
   The process used to generate a unique KDC per cryptographic module takes place outside of the crypto module.

2. **KSC**: Initial Issuer Session Transport Keyset: Set of two transient Triple DES Keys (called KSC_{ENC} KSC_{MAC}) of 16 bytes each. KSC_{ENC} is used for Secure Channel Encryption, and KSC_{MAC} is used for Secure Channel MAC verification.

KDC_{ENC} and KDC_{MAC} are used to derive KSC_{ENC} and KSC_{MAC} keys that are used to authenticate the secure sessions with the card Manager.

KDC_{KEK} does not derive any keys but is used directly to wrap the CO CDK key set and the User ADK key set when they are entered into the module for the first time. As a result this triple DES KDC can only be used to wrap/unwrap other triple DES keys of the same size.

14.2.2 Crypto-Officer keys in Card Manager

1. **CDK**: Crypto-Officer Keyset: Set of three Triple DES Keys (called CDK_{ENC} CDK_{MAC} and CDK_{KEK}) of 16 bytes each. The first two, CDK_{ENC} and $CDK_{MAC,}$ are only used to generate Secure Channel session keys (CSK_{ENC} and CSK_{MAC}) during the initiation of a Global Platform Secure Channel, and the last one, CDK_{KEK} is used as a key transport key within the secure channel to wrap only other triple DES keys of the same size.
   The process used to generate a unique CDK per cryptographic module takes place outside of the crypto module.
2. **CSK**: Crypto-Officer Session Keyset: Set of two transient Triple DES Keys (called CSK_{ENC} and CSK_{MAC}) of 16 bytes each. CSK_{ENC} is used for Secure Channel Encryption, and CSK_{MAC} is used for Secure Channel MAC verification.
3. **K_{TOKEN}**: Key Token: Public RSA Key (1024 bits) used to verify the tokens included in Delegated Management commands that embed the signature of these commands. This key may or may not be loaded into the module. It is an added feature and is not intended to satisfy any of the FIPS 140-2 requirements for applet loading.
4. **$K_{RECEIPT}$:** Key Receipt: Triple DES Key (16 bytes) used to compute a receipt on Delegated Management Commands. See Delegated Management in section 14.3.2. This key may or may not be loaded into the module. It is an added feature and is not intended to satisfy any of the FIPS 140-2 requirements for applet loading.

14.2.3 User/Applet Provider Keys in Security Domains

1. **ADK**: Applet Provider Keyset: Set of three Triple DES Keys (called ADK_{ENC} ADK_{MAC} and ADK_{KEK}) of 16 bytes each. The first two, ADK_{ENC} and $ADK_{MAC,}$ are only used to generate Secure Channel session keys (ASK_{ENC} and ASK_{MAC}) during the initiation of a Global Platform Secure Channel, and the last one, ADK_{KEK} is used as a key transport key within the secure channel to wrap only other triple DES keys of the same size. This keyset is present in both type of Security Domain, Security Domain with Delegated Management, and Security Domain with DAP Verification. The process used to generate a unique ADK per cryptographic module takes place in the cryptographic HSM outside of the crypto module.
2. **ASK**: Applet Provider Session Keyset: Set of two transient Triple DES Keys (called ASK_{ENC} and ASK_{MAC}) of 16 bytes each. ASK_{ENC} is used for Secure Channel Authentication and optionally Encryption, and ASK_{MAC} is used for Secure Channel MAC verification.
3. **K_{DAP}**: Key DAP: Public RSA Key (1024 bits) used to verify the DAP on an application code to be loaded into the module and authorize or not its loading. (See section 14.3.3 on DAP verification). This key may or may not be loaded into the module. It is an added feature and is not intended to satisfy any of the FIPS 140-2 requirements for applet loading. This key is present only in Security Domain with DAP

Verification. More information on how this key is used can be found in section 10.2.3, DAP VERIFICATION.

14.2.4 Keys Exchange

The following key exchange takes place with the Cryptographic Officer and with the User/Applet provider prior to the module being initialized by Oberthur.

The values of the root secrets used to retrieve a module unique CDK, (with optionally $\mathbf{K}_{RECEIPT}$) and ADK are securely exchanged between Oberthur production HSM and respectively the Cryptographic Officer HSM and the User/applet provider HSM using a well defined and highly secure key ceremony described in a separate document.

The values of the RSA public Keys $\mathbf{K}_{TOKEN}$ and $\mathbf{K}_{DAP}$, are provided respectively by the Cryptographic Officer and the User/applet provider using a method that guarantees the integrity but not necessarily the confidentiality of the transmission.

In this FIPS configuration, the CSK and the ASK are the only keys that are not loaded but generated automatically by the module whenever needed. The keys are generated via the Open Platform Card Specification Secure Session Key Generation Process that was approved by NIST/CSE during the first Java based smart card validation. All Java-based smart cards use this process to generate sessions keys.

14.2.5 Key Loading

During the card manufacturing and initialization process, an initial set of Open Platform Keys called KDC is securely loaded into the Card Manager (Crypto-Officer) Security Domain. This key set is generated by a derivation process using a master secret key called KMC and card specific information such as chip serial number.

The KDC keyset is used to open a Secure Channel that will protect the loading of the initial value of Crypto-Officer and User Keys (except for the transient session keys ASK and CSK that are not loaded but generated automatically by the module whenever needed).

Crypto-Officer and User Key Loading is done using an authentication followed by a PUT3KEY or PUT PUBLIC KEY command depending on the type of key being loaded. Keys are valid until replaced

Both the Crypto-Officer and the User/Applet provider can replace their own keys at anytime during the active life of the module or whenever they feel a key may be compromised. This is done using an authentication with the current keyset (CDK or ADK) followed by a Put Key command with “Key update” as parameter. Depending on the key to replace, the PutKey command is actually a PUT3KEY or a PUT PUBLIC KEY command. The new value is loaded into the card encrypted with the old key-set value using the TDES algorithm.

14.2.6 EEPROM encryption Key

A higher level of protection against attack is provided through the encryption with TDES_CBC of the EEPROM content. This is done in an automated and transparent way by the module operating system using a special 3DES key called LSK. LSK is a 16 byte TDES key. The LSK is entered during wafer manufacturing by the silicon vendor.

14.3 Card Cryptographic Functions

The purpose of the cryptographic module is to provide a FIPS approved platform for applets that may in turn provide cryptographic services to end-user applications. The keys represent the identity of the roles involved in controlling the module. A variety of FIPS 140-2 validated algorithms are used in the ID-One Cosmo 32 v5 to provide cryptographic services.

Some of these cryptographic services are made available only to applets and through Java APIs. Since the module described in this security policy does not include any instantiated applets other than the Card Manager and Security Domains, security services not used by either the Card Manager or by the Security Domain are not available to any of the current operator of the module. They are however listed here in italic font to inform applet developers of all cryptographic services built into the module.

The cryptographic functions provided by the ID-One Cosmo 32 v5 include:

- **DES [Certificate # 286]:** DES functions are available for legacy systems (Transitional phase only valid until May 19, 2007).
- **TDES, (2 keys TDES) [Certificate # 303]:** The TDES (CBC mode) algorithm is used:
    - For authenticating the Crypto-Officer (EXTERNAL AUTH command)
    - For encrypting data flow from the off module to the on-module environment. The reverse direction is not encrypted; i.e. the status words returned in response to an APDU are not encrypted.
    - As a TDESMAC to authenticate the originator and to the verification the integrity of the message.

  TDES is also used to sign receipts from Delegated Management.

  TDES functions (CBC and ECB) are also provided as services to applets, through Java APIs. These services are not available to any of the current operators of the module.
- ***Full TDES, (3 keys TDES)*** **[Certificate # 303]:** *Full TDES functions are provided as services to applets, through Java APIs.*
- ***SHA-1*** **[Certificate # 290]:** *The SHA-1 function is provided as a service through Java APIs to applets.*
- ***MD5:*** *The MD5 function is provided as a service through Java APIs to applets. It will not be used by FIPS validated applets in FIPS mode.*
- **RSA (up to 2048 bit keys)[7] [Certificate # 42]:** RSA functions are provided as services to Card Manager (Delegated Management) and to Security domain (DAP verification) as well as to applets through Java APIs. The applet shall use RSA only for “key wrapping” or “signature”. This will be checked during the applet FIPS validation. RSA can be configured to be compliant with PKCS#1, PSS, ISO/IEC 9796 and ANSI X9.31. The minimum RSA key size for the module to operate in FIPS mode is 1024 bits. The RSA algorithm implemented in the module supports RSA key sizes up to 2048 bits by 64 bit increments. DAP and Delegated Management use PKCS#1 implementation of the RSA signature algorithm with a 1024 bit key.

DES, TDES, Full TDES, RSA, SHA-1 and MD5 algorithms are provided as services to applets that may be loaded onto the ID-One Cosmo 32 v5 module. These algorithms, except MD5 that is outside the list of approved algorithms, shall be used only in a FIPS approved mode of operation. If the MD5 algorithm is used, the module is no longer in an approved mode of operation.

[7] In this FIPS configuration, without an application (applet) loaded, the only available service which utilizes the RSA algorithm is one that uses an RSA public key for DAP verification (see section 10.2.3, DAP VERIFICATION). As a result, the use of RSA for encryption, decryption, key wrapping and unwrapping, is not available in this module.

14.3.1 Random Number Generators [FIPS Certificate # 99]

The cryptographic module offers the services of a FIPS 140-2 approved DRNG (Deterministic Random Number Generator). The random generation algorithm has been certified to be compliant with the FIPS PUB 186-2 standard [Certificate # 99].

The cryptographic module also offers the services of a hardware based NDRNG (Non Deterministic Random Number Generator), which can be used to generate a seed to feed the DRNG and increase its quality.

14.3.2 Delegated Management

The design of the Oberthur ID-One Cosmo 32 v5 module takes into account the possibility that the Card Issuer (Cryptographic Officer) may not necessarily want to manage all Card Content changes, especially when the Card Content does not belong to the Card Issuer. The concept of Delegated Management defined by Global Platform gives the Card Issuer the possibility of empowering partnered Application Providers the ability to initiate approved and pre-authorized Card Content changes (loading, installation, extradition[8] or deletion). This approval, which is central to the concept of Delegated Management, ensures that only Card Content changes that the Card Issuer (Cryptographic Officer) has authorized will be accepted and processed by the module. This delegation of control in the Card Content changes gives the Application Provider more flexibility in managing its Application.

The Security Domain with the delegated management privilege allows making:

- Delegated loading (requires a pre-authorization)
- Delegated installation (requires a pre-authorization)
- Delegated extradition (requires a pre-authorization)
- Delegated deletion (no pre-authorization required)

The Delegated Management is based on the use of Token. A token is a cryptographic value provided by a Card Issuer (Cryptographic Officer) as proof that a specific Delegated Management operation has been authorized.

Delegated Management Tokens are RSA PKCS1 signatures of one or more Delegated Management functions and a hash of associated data (loading application code, installing Applications and extraditing Applications) generated by the Card Issuer (Cryptographic Officer) outside of the crypto module and transmitted to a user with Delegated Management privilege. The public RSA key $\mathbf{K}_{\text{TOKEN}}$, associated with the Crypto-Officer token signature private RSA key, must be present in the Card Manager.

When the User wants to perform the pre-authorized function, it appends to the function's data transmitted through a secure channel with its Security Domain inside the ID-One Cosmo 32 v5 platform the associated token. The User security domain will then decrypt and verify the secure channel communication using its ASK. The function and its associated Token are then automatically transmitted to the Crypto-Officer Card Manager for token verification using the Card Manager $\mathbf{K}_{\text{TOKEN}}$ Public RSA key. If the signature is verified, the function is authorized to complete. Otherwise, it is aborted and cleared for memory.

The Card Issuer's security policy may require the generation of Receipts for Delegated Management operations. A Receipt is a cryptographic value (DES signature on the receipt data) generated by the Card Manager $\mathbf{K}_{\text{RECEIPT}}$ key to provide confirmation from the card that a successful card content management function has occurred through the delegated installation process. The Install Receipt is comprised of data related to the delegated card content management function including Card Unique Data generated by the Card Manager. The card manager also keeps track of a Confirmation Counter value that is incremented when generating each Receipt.

[8] Application Extradition allows an Application that is already associated with a Security Domain to be extradited and associated with another Security Domain

The receipt is computed by the Card Manager using the $\mathbf{K}_{\mathrm{RECEIPT}}$, an ICV of binary zeroes and the signature method described in Global Platform 2.1.1, Appendix B.1.2.2 - Single DES Plus Final Triple DES MAC.

14.3.3 DAP Verification

If the Application Provider does not have a Security Domain capable of Delegated Management to load application code to the card, it may rely on the loading services of the Card Issuer (Cryptographic Officer) and require a check of application code integrity and authenticity before the application code is loaded by the Crypto-Officer. Likewise, a Controlling Authority may mandate a check of application code integrity and authenticity before the application code is loaded, installed and made available to the Cardholder by the crypto-Officer or by a User with Delegated Management. The DAP Verification privilege for a User Security Domain provides this service on behalf of an Application Provider. The mandated DAP Verification privilege provides this service on behalf of a Controlling Authority.

The way it works is as follows: The user first computes a SHA-1 message digest of the application that is to be subsequently loaded into the module. He then uses his DAP RSA private key (matching the public key $\mathbf{K}_{\mathrm{DAP}}$ in the user security domain) to sign the previously calculated hash. The result, called DAP, is sent to the personalization entity together with the application code itself. When the application must be loaded into the card, the User Security Domain with DAP verification uses its DAP public key $\mathbf{K}_{\mathrm{DAP}}$ to check the DAP signature. The application code can be loaded into the module only if the verification succeeds.

15 EMI/EMC

The cryptographic module meets the Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC) requirements specified by United States Standards 47 CFR Part 15, Subpart B: “Unintentional Radiators, Digital Devices, Class B”.

It is also in compliance with the electromagnetic compatibility requirements defined in European Standard EN 55022, Class B: “Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment”.

16 Self Tests

16.1 Power Up Self Tests

Each time the ID-One Cosmo 32 v5 module is powered by a reader (contact or contactless), a “reset” signal is sent from the reader to the module. The module then performs a series of GO/NO-GO tests to validate that the cryptographic module is in good working order before it answers to the reset signal with an Answer To Reset (ATR) packet of information as specified by ISO/IEC 7816 (for contact mode) or with an Answer To Select (ATS) as defined in ISO/IEC 14443 for contactless mode.

The Power-up self tests include:

- EEPROM integrity check for:
  - System Data
  - optional codes (firmware extensions), if any
  - uploaded application packages (Executable Load files), if any
- Cryptographic Known Answer Tests for
  - DES – Encryption and decryption in ECB mode
  - Triple DES – Encryption and decryption in CBC mode
  - SHA-1 Hashing
  - RSA signature generation and signature verification
  - RSA encryption and decryption
  - Deterministic Random Number Generator (DRNG)
- Critical Function Tests
  - CRC-16 KAT
  - RAM functional test
  - Sensor bit test
  - Audit log scan
  - Resident applet life cycle

Additional tests to protect against new types of attacks such as SPA, DPA, “flash gun”, etc, are also performed at this stage.

No data of any type (except error status) is transmitted from the cryptographic module to the reading device while the self-tests are being performed.

If any of the above tests fail, the card will enter en error state in which further APDU’s are not processed. Depending on the test that fails, the module may return the ATR/ATS with an error status before becoming mute.

More details about all the power-up self-tests and their implementation are provided in a separate confidential document.

16.2 Conditional Tests

RSA Key generation: After generating an RSA key pair, the module performs a double pair wise consistency check to validate that the generated key pair is correct for both signature/verification and encryption/decryption. Description of the implementation of this test is provided in a separate document.

Random Number Generator: Continuous testing is performed on every output of the Random Number Generator. Checks on the non-deterministic (Hardware) component are made on 16 bits and checks on deterministic part (FIPS approved) are made on 160 bits. Description of the implementation of this test is provided in a separate document.

Credentials: Keys and PINs: Each time a credential is used, whether a TDES, RSA key or a PIN, a signature of the credential value is computed and compared to the expected signature stored in the EEPROM along with the object itself. If these values are the same, the test is successful and the object can be used safely. If the values are different, the object cannot be used. The problem is recorded into a special "audit file" and a security exception is thrown, causing the execution to abort. Description of the implementation of this test is provided in a separate document.

Software (Applet) load tests: A TDES CBC MAC on the applet executable load file is verified each time an applet is loaded onto the cryptographic module since applet loading always take place within a Secure Channel. This is done as part of the secure channel MAC verification using CSK_{MAC} described in 14.2.2. Crypto-Officer keys in Card Manager. In addition, if a Security Domain with mandated DAP verification is installed, every executable load file loaded into the cryptographic module has to provide a DAP value. This is an RSA PKCS#1 signature (1024 bit) verification of the SHA-1 message digest of the complete load file data block that enables the Applet Provider Security Domain to check that the applet has been correctly authorized. The RSA cryptographic key used for this verification is the $\mathbf{K}_{DAP}$ defined in 14.2.3 User/Applet Provider Keys in Security Domains.

If TDES MAC or DAP verification fails, package load is terminated and the module built-in garbage collector clean the EEPROM of any traces of the aborted download.

Description of the implementation of this test is provided in Global platform Specifications.

17 Security Rules

17.1 Identification & Authentication Security Rules

The module implements specific methods for identifying and authenticating the different roles. The implementation consists of the binding of an Identity-based Access Control Rule to each service.

17.1.1 User Identification and Authentication

The operator who wishes to authenticate into the User/Applet Provider role must first identify him/herself by providing both an application identifier and a Key Set ID. The authentication is then done by proving the possession of the particular keyset identified in the identification phase. This Key Set is composed of 3 TDES keys. One key is used to encrypt the command data, one key to authenticate the user, and the third key is used to encrypt keys transported within the APDU command. This is the same process as the Crypto-Officer authentication (Initialize Update & External Authenticate commands) but it uses the TDES keys of the Applet Provider Security Domains.

17.1.2 Cryptographic Officer Identification &Authentication

The operator that wishes to authenticate into the Cryptographic Officer Provider role must first identify him/herself by providing both information to uniquely select the Card Manager, and a Key Set ID. The authentication is then done by proving the possession of the particular keyset identified in the identification phase. This Key Set is composed of 3 TDES keys. One key is used to encrypt the command data, one key to authenticate the user, and the third key is used to encrypt keys transported within the APDU command. This is the same process as the User authentication (Initialize Update & External Authenticate commands).

17.2 Applet Loading Security Rules

Applets can only be loaded through a secure channel; i.e. they pass from the off-module to the on-module environment in a MACed form. An additional applet encryption is also available as an option. To activate that option, the Cryptographic Officer would have to request the encryption option when opening the secure channel with the module. The applet code would then not only be MACed, using CSK_{MAC}, but also encrypted using CSK_{ENC}. Both keys are described in section 14.2.2 Manager. In the ID-One Cosmo 32 v5 platform, the applet is always loaded by the Issuer (Cryptographic Officer). The optional mechanism designated as “DAP” in GP 2.1.1 enables the applet provider to check, independently of the Issuer (Cryptographic Officer), that his applet has been correctly loaded. This check is done by verifying an RSA PKCS#1 signature on the Hash of the applet code being loaded. This process is described in detail in the GP 2.1.1 document. See section 2 Applicable Documents.

For the ID-One Cosmo 32 v5 to run in a validated FIPS 140-2 Level 3 mode of operation, all applet instances must be validated to the same level. Although any applet can be loaded during post issuance, it is the responsibility of the Cryptographic Officer to insure that only FIPS 140-2 Level 3 validated instances are created. Instantiation of non-validated applets within the FIPS 140-2 validated cryptographic module, or instantiation of a FIPS 140-2 validated applet with a different security level, will invalidate the original validation.

FIPS 140-2 Level 3 validated applets may be loaded and instantiated at post issuance.

17.3 Key Management Security Policy

17.3.1 Cryptographic key generation

TDES Session key derivation for Secure Channel Opening, conforming to Open Platform Card Specification v2.1 (SCP01 or SCP02) using FIPS186-2 approved ANSI X9.31 DRNG.

RSA key pair generations (up to 2048 bit key length) with strong prime numbers (ANSI X9.31) using FIPS140-2 approved DRNG. Both standard RSA key and RSA Chinese Remainder Keys can be generated. This cryptographic service is made available through Java APIs only.

17.3.2 Cryptographic key entry

Keys shall always be input in encrypted format, using the Put Key (TDES or Public) command within a secure channel. During this process, the keys are encrypted using the Key Encryption Key and optionally the encryption session key of the secure channel.

Keys can never be output by the module.

17.3.3 Cryptographic key storage

The Keys are structured to contain the following parameters:

- Key set version
- Key Index, which is the ID of the key,
- Algo ID, which determines which algorithm to be used,
- Integrity Mechanisms.

The cryptographic key storage integrity mechanism is described in a separate confidential document called Self Test Description.

17.3.4 Key Destruction

The ID-One Cosmo 32 v5 destroys cryptographic keys by reloading another key-set with the same version number for Crypto-Officer Keys and User/Application Provider Keys, using the **PUT TDES KEY** or **PUT PUBLIC KEY** command.

User/Application Provider Keys can also be zeroized by deleting the Security Domain that hosts the keys, using the **DELETE** command.

Closing of the secure channel has also the effect of zeroizing the associated session keys stored in RAM memory.

Key zeroisation is achieved by the Oberthur Garbage Collector that overwrites with binary zeros the deleted key value in its memory zone (whether in RAM or in EEPROM).

18 Mitigation of Other Attacks Policy

18.1 Power Analysis (SPA/DPA)

Power analysis attacks use information gathered from non-invasive measurements to cryptanalyse and extract keys from tamper resistant devices.

Simple Power Analysis (SPA) attacks use direct observation of a device's power consumption. Because power consumption often varies significantly with computations performed by the crypto module, SPA observations can identify sensitive computational processes, reveal the presence of cryptographic sub-routines, and significantly accelerate reverse engineering.

Differential Power Analysis (DPA) attacks use statistical analysis and error correction techniques to extract information leaked across multiple operations. This aggregation of data allows extremely small differences in power consumption to be isolated, including effects that are many orders of magnitude smaller than "noise".

The ID-One Cosmo 32 v5 has been designed to mitigate both Simple Power Analysis (SPA) and Differential Power Analysis (DPA).

The module includes protections against SPA and DPA attacks for all embedded cryptographic algorithms involving secret elements. The chip protection level was evaluated against state-of-the art attacks (at the time of design).

The cryptographic module mitigates Simple Power Analysis (SPA) and Differential Power Analysis (DPA) attacks using a combination of hardware and software design that makes differentiation of key values impractical by equalizing or scrambling current consumption of the card during algorithm cryptographic computation.

Based on the algorithm used, the defense mechanisms vary, as the internal hardware implementations of these algorithms do not use the same underlying hardware.

18.2 Timing Analysis

Timing attacks are non invasive attacks that relies on the variation in computation time required for the microprocessor to perform its secret calculation.

All cryptographic algorithms as well as Java Card API comparison functions offered by the chip are designed to be protected against Timing Analysis.

This is done by enforcing the fact that any sensitive operation is achieved in a constant time regardless of the value of keys or data involved.

18.3 Fault Induction

This type of attack is based on the theoretical possibility of flipping some random bits of the secret key, stored in RAM or EEPROM, before or during the computation done by the module. (Bellcore attack). Another fault induction attack is to induce decoding error during the execution of one instruction.

The ID-One Cosmo 32 v5 includes a combination of software and hardware protections in order for the chip not to operate in extreme conditions that may cause processing errors that could lead to revealing the values of cryptographic keys or secret elements. Extreme Conditions refer to abnormal temperature, external power supply and external clock supply.

In addition, every keys and PINs are protected by a signature that is checked prior to every use of the keys or PINS. See section 16.2 Conditional Tests

18.4 Flash Gun

The ID-One Cosmo 32 v5 includes a combination of software and hardware protections in order to detect "Flash Gun" type of attacks and abort any current processing before becoming mute.

19 API Services for future validations

The module described in this security policy, which is validated to FIPS 140-2, does not include any instantiated applets other than the Card Manager and Security Domains. API services provided by the module are therefore not used in this configuration. They are however listed here to inform applet developers of additional services made available to them by the ID-One Cosmo 32 v5 platform.

The ID-One Cosmo 32 v5 supports the following API:

- JavaCard 2.2 API
- Global Platform 2.1.1 API
- Org.GlobalPlatfrom API (version 2.1.1)
- Visa Open Platform 2.0.1' API
- ISO 7816 File System API
- BIO API

Details on these API services can be found in the associated reference documents listed in Section 2 Applicable Documents.

Among them, the following contain cryptographic services:

19.1 Key Generation:

- **ALG_RSA_CRT**: this API generates a pair of RSA_CRT keys from 256 to 2048 bits by steps of 64 bits using strong primes as per ANSI X9.31.

19.2 Message Digest:

- **ALG_MD5**[9] Message Digest algorithm MD5.
- **ALG_SHA:** Message Digest algorithm SHA.

[9] The MD5 algorithm is not a FIPS approved algorithm and is provided only for backward compatibility purposes. New applications should refrain from using this algorithm.

19.3 Bulk Encryption/Decryption:

- **ALG_DES_CBC_NOPAD**: Cipher algorithm ALG_DES_CBC_NOPAD provides a cipher using DES in CBC mode. This algorithm uses outer CBC for triple DES. This algorithm does not pad input data.
- **ALG_DES_CBC_ISO9797_M1**: Cipher algorithm ALG_DES_CBC_ISO9797_M1 provides a cipher using DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 1 scheme.
- **ALG_DES_CBC_ISO9797_M2**: Cipher algorithm ALG_DES_CBC_ISO9797_M2 provides a cipher using DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 2 (ISO 7816-4, EMV'96) scheme.
- **ALG_DES_ECB_NOPAD**: Cipher algorithm ALG_DES_ECB_NOPAD provides a cipher using DES in ECB mode. This algorithm does not pad input data.
- **ALG_DES_ECB_ISO9797_M1**: Cipher algorithm ALG_DES_ECB_ISO9797_M1 provides a cipher using DES in ECB mode. Input data is padded according to the ISO 9797 method 1 scheme.
- **ALG_DES_ECB_ISO9797_M2**: Cipher algorithm ALG_DES_ECB_ISO9797_M2 provides a cipher using DES in ECB mode. Input data is padded according to the ISO 9797 method 2 (ISO 7816-4, EMV'96) scheme.
- **ALG_RSA_NOPAD**: Cipher algorithm ALG_RSA_NOPAD provides a cipher using RSA. This algorithm does not pad input data.
- **ALG_RSA_PKCS1**: Cipher algorithm ALG_RSA_PKCS1 provides a cipher using RSA. Input data is padded according to the PKCS#1 (v1.5) scheme.
- **ALG_RSA_PKCS1_OAEP**: Cipher algorithm ALG_RSA_PKCS1_OAEP provides a cipher using RSA. Input data is padded according to the PKCS#1-OAEP scheme (IEEE 1361-2000).

19.4 Signature and Verification:

- **ALG_DES_MAC4_NOPAD**: Signature algorithm ALG_DES_MAC4_NOPAD generates a 4 byte MAC (most significant 4 bytes of encrypted block) using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. This algorithm does not pad input data.
- **ALG_DES_MAC4_ISO9797_M1**: Signature algorithm ALG_DES_MAC4_ISO9797_M1 generates a 4 byte MAC (most significant 4 bytes of encrypted block) using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 1 scheme.
- **ALG_DES_MAC4_ISO9797_M2**: Signature algorithm ALG_DES_MAC4_ISO9797_M2 generates a 4 byte MAC (most significant 4 bytes of encrypted block) using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 2 (ISO 7816-4, EMV'96) scheme.
- **ALG_DES_MAC4_ISO9797_1_M2_ALG3**: Signature algorithm ALG_DES_MAC4_ISO9797_1_M2_ALG3 generates a 4 byte MAC using a 2-key DES3 key according to ISO9797-1 MAC algorithm 3 with method 2 (also EMV'96, EMV'2000). Input data is padded using method 2 and the data is processed as described in MAC Algorithm 3 of the ISO 9797-1 specification.
- **ALG_DES_MAC8_NOPAD**: Signature algorithm ALG_DES_MAC_8_NOPAD generates an 8-byte MAC using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. This algorithm does not pad input data.

- **ALG_DES_MAC8_ISO9797_M1**: Signature algorithm ALG_DES_MAC8_ISO9797_M1 generates an 8-byte MAC using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 1 scheme.
- **ALG_DES_MAC8_ISO9797_M2**: Signature algorithm ALG_DES_MAC8_ISO9797_M2 generates an 8-byte MAC using DES or triple DES in CBC mode. This algorithm uses outer CBC for triple DES. Input data is padded according to the ISO 9797 method 2 (ISO 7816-4, EMV'96) scheme.
- **ALG_DES_MAC8_ISO9797_1_M2_ALG3**: Signature algorithm ALG_DES_MAC8_ISO9797_1_M2_ALG3 generates an 8-byte MAC using a 2-key DES3 key according to ISO9797-1 MAC algorithm 3 with method 2 (also EMV'96, EMV'2000). Input data is padded using method 2 and the data is processed as described in MAC Algorithm 3 of the ISO 9797-1 specification.
- **ALG_RSA_SHA_ISO9796**: Signature algorithm ALG_RSA_SHA_ISO9796 encrypts the 20-byte SHA digest using RSA. The digest is padded according to the ISO 9796-2 (EMV'96, EMV'2000) scheme.
- **ALG_RSA_SHA_PKCS1**: Signature algorithm ALG_RSA_SHA_PKCS1 encrypts the 20-byte SHA digest using RSA. The digest is padded according to the PKCS#1 (v1.5) scheme.
- **ALG_RSA_SHA_PKCS1_PSS**: Signature algorithm ALG_RSA_SHA_PKCS1_PSS encrypts the 20 byte SHA-1 digest using RSA. The digest is padded according to the PKCS#1-PSS scheme (IEEE 1363-2000).
- **ALG_RSA_MD5_PKCS1**: Signature algorithm ALG_RSA_MD5_PKCS1 encrypts the 16-byte MD5 digest using RSA. The digest is padded according to the PKCS#1 (v1.5) scheme.
- **ALG_RSA_MD5_PKCS1_PSS**: Signature algorithm ALG_RSA_MD5_PKCS1_PSS encrypts the 16 byte MD5 digest using RSA. The digest is padded according to the PKCS#1-PSS scheme (IEEE 1363-2000).

19.5 Random Numbers Generation:

- **ALG_PSEUDO_RANDOM**: Utility pseudo random number generation algorithms. This API uses the FIPS 140-2 approved DRNG with an externally supplied seed. The DRNG is compliant with FIPS PUB 186-2 and specified in section 14.3.1 Random Number Generators [FIPS Certificate # 99].
- **ALG_SECURE_RANDOM**: Cryptographically secure random number generation algorithms. This API uses the FIPS 140-2 approved DRNG using a seed provided by the hardware based NDRNG. The DRNG is compliant with FIPS PUB 186-2 and specified in section 14.3.1 Random Number Generators [FIPS Certificate # 99].

The above algorithms are provided as services to applets that may be loaded onto the module. These algorithms shall only be used in FIPS approved mode of operation. This will be checked during the applet's validation. Since the MD5 algorithm is not a FIPS approved algorithm, it cannot be used on a FIPS approved mode of operation.

20 Security Policy Check List Tables

20.1 Roles and required Identification and Authentication

Role	Type of Authentication	Authentication Data
Crypto-Officer	TDES Authentication	TDES Keys (Crypto-Officer Security Domain)
User/Applet Provider	TDES Authentication	TDES Keys (User/Applet Provider Security Domain)

20.2 Strength of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
TDES Authentication	The probability that a random authentication attempt succeeds is less than 1 in 1,000,000
RSA Authentication	The probability that a random authentication attempt succeeds is less than 1 in 1,000,000

20.3 Services Authorized for Roles

Role	Authorized Services
Crypto-Officer	All Crypto-Officer Services are listed in section 10.2.
User/Applet Provider	All User/Applet Provider Services are listed in section 10.2.

20.4 Access Right within Services

CSP	Service	Role	Type of Access
CDK_{ENC} CDK_{MAC} and CDK_{KEK}	PUT TDES KEY command	CO	Write
CDK_{ENC} and CDK_{MAC}	INITIALIZE UPDATE & EXTERNAL AUTH	CO	Read & Execute
CDK_{KEK}	TDES Key Encryption (during key loading)	CO	Write
CDK_{ENC} & CDK_{MAC}	Message Integrity & Encryption	CO	Read & Execute
CDK_{MAC}	Message Integrity Only	CO	Read & Execute
$K_{RECEIPT}$	Delegate Management	CO	Read & Execute
K_{TOKEN}	Delegate Management	CO	Read & Execute
Global PIN	PIN UNBLOCK command	CO	Write
K_{DAP}	PUT RSA KEY command	CO	Write
ADK_{ENC} ADK_{MAC} and ADK_{KEK}	PUT TDES KEY command	User	Write
ADK_{ENC} and ADK_{MAC}	INITIALIZE UPDATE & EXTERNAL AUTH	User	Read & Execute
ADK_{KEK}	TDES Key Encryption (during key loading)	User	Write
ADK_{ENC} & ADK_{MAC}	Message Integrity & Encryption	User	Read & Execute
ADK_{MAC}	Message Integrity Only	User	Read & Execute
K_{DAP}	RSA PKCS#1 Signature Verification by Applet Provider Security Domain	User	Read & Execute
LSK	EEPROM encryption key	All	Read & Execute

20.5 Mitigation of Other Attacks

Other Attacks	Mitigation Mechanism	Specific Limitations
Simple Power Analysis	Counter Measures against SPA	N/A
Differential Power Analysis	Counter Measures against DPA	N/A
Timing Analysis	Counter Measures against TA	N/A
Fault Induction	Counter Measures against FI	N/A
Flash Gun	Counter Measures against FG	N/A