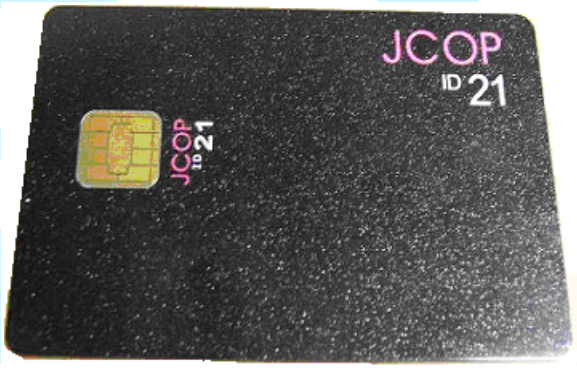
FIPS140-2 Security Policy for the JCOP21id 32K JavaCard Platform



FIPS140-2 Security Policy for the JCOP21id JavaCard Platform

November 2003

Revision: 1. 12

NON CONFIDENTIAL

Status: Released

First Edition (November 2003)
This edition applies to the First Edition of the IBM BlueZ – FIPS140-2 Security Policy for the JCOP21id JavaCard Platform with PKCS#15 Applet and to all subsequent versions until otherwise indicated in new editions. IBM welcomes your comments on this publication. Please address them to: javacard@zurich.ibm.com. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

1. Document Information

1.1. Table of Contents

1.2. Document Scope

This document describes the services that the IBM JCOP21id 32K provides to a population of security officers, and the security policy governing access to those services. Included is a description of the basic security requirements for the 'JCOP21id 32K' card and a qualitative description of how each of the security requirements is achieved. This document is prepared as part of the JCOP21id 32K (firmware version: mask 20, hardware version: P8WE5033AEV/034188i, Applet version 1.0) validation as a single-chip, level 3, FIPS 140-2 compliant product.

1.3. Applicable documents

1.3.1. Hardware

P8WE5017 Secure 8-bit Smart Card Controller: Short Form Specification Revision 1.1 June 2001, Phillips Semiconductors
URL: http://www.semiconductors.philips.com/acrobat/other/identification/sfs053811.pdf

1.3.2. Smart Cards

ISO, Information Technology - Identification cards - Integrated circuit(s) cards with contacts

Part 1: Physical Characteristics, ISO/IEC 7816-1

Part 2: Dimensions and location of the contacts, ISO/IEC 7816-2

Part 3: Electronic signals and transmission protocols, ISO/IEC 7816-3

Part 4: Inter-Industry commands for interchange, ISO/IEC 7816-4

Part 5: Numbering system and registration procedure for application identifiers, ISO/IEC 7816-5

Part 6: Inter-industry data elements, ISO/IEC 7816-6

Part 8: Security related Inter-industry commands, ISO/IEC FDIS 7816-8 (draft)

1.3.3. FINEID

FINEID S1 - Electronic Identity Application, v1.1
FINEID S4-1 - Implementation Profile 1, v1.1

1.3.4. Common Criteria

The Common Criteria for Information Technology Security Evaluation Version 2.1, August 1999 (ISO 15408:1999)
The Open Platform Protection Profile, Version 0.5.0.1 issued May 2000 www.visa.com
Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), DOD 5200.28-STD, December 1985.
The Smart Card Security User Group Smart Card Protection Profile, Draft Version 2.0, 1 May 2000 http://csrc.nist.gov/cc/sc/sclist.htm
Protection Profile 9806 - Smartcard Integrated Circuit (revision of PP 9704 - Smartcard Integrated Circuit), Protection Profile 9810 - Smartcard Embedded Software, Protection Profile 9911 - Smart Card Integrated Circuit with Embedded Software (supersedes PP9809 - Smart Card Integrated Circuit with Embedded Software), http://www.eurosmart.com and http://www.scssi.gouv.fr

1.3.5. Cryptography

RSA Laboratories PKCS #15 v1.0: Cryptographic Token Information Format Standard – April 23, 1999
RSA Laboratories PKCS#15 v1.0 Amendment 1 Draft #1 - October 20, 1999

1.3.6. JavaCard

JavaCard™ 2.1.1 Specifications, May 2000

1.3.7. Open Platform

Open Platform Card Specification, Version 2.1 issued June 2001,

1.4. Glossary of Terms

Term	Meaning
ACR	Access Control Rule
APDU	Application Protocol Data Unit
API	Application Programming Interface
C.O.	Cryptographic Officer
JCRE	Java Card Run-Time Environment
OP	Open Platform
MAC	Message Authentication Code
PIN	Personal Identification Number
SRDI	Security Relevant Data Item
EEPROM	Electrically Erasable Programmable Read Only Memory.
CAD	Card Acceptance Device
ICC	Integrated Circuit Chip
RAM	Random Access Memory
ROM	Read Only Memory.
JCVM	Java Card Virtual Machine
RFU	Reserved for Future Use

2. JCOP21id 32K

2.1. Module Description

JCOP21id 32K is a smart card or integrated circuit card (ICC), which is a computer chip embedded into a carrier. The chip is a semiconductor (silicon) integrated circuit (IC) manufactured by Philips Semiconductors – part number P8WE5033AEV/034188i. The chip is fabricated in an advanced CMOS process that involves repeatedly masking and doping the surface of a silicon substrate to form transistors, followed by patterning metal connections, and applying a protective overcoat. This process eventually yields a design typically comprising several hundred thousand transistors, arranged in an area less than 25 square millimeters. The CPU of the Philips chip is a derivation of the 80C51 family and has the same instruction set. The design consists of a central processing unit, various coprocessors, input and output lines, and volatile and non-volatile memory. The EEPROM can be addressed as data memory as well as program memory.

For further hardware details see the document "P8WE5033 Secure 8-bit Smart Card Controller: Short Form Specification" Revision 1.1 June 2001, Phillips Semiconductors

The chip is designed to be secure with appropriate use of the technological properties of the materials and processes used. A part of the manufacturing process is the inclusion of operating software (OS). This developer specific code created at IBM is contained in one of the numerous masks used during manufacture, referred to in this document as the ROM mask. Parts of the OS may, however, be stored in programmable non-volatile memory (e.g., outside of the ROM mask). The IC itself is packaged. The current predominant method is die bonding in a module. A module consists of a small board on which the IC is seated. Wire bonds are connected from a subset of the IC's input/output (I/O) pads to the carrier, which has contacts on its reverse side. The chip is encapsulated in a hard opaque protective material and the module is adhesively embedded into a pre-milled hole in the plastic card.

The 'JCOP21id 32K' smart card contains an implementation of the Open Platform (OP) Version 2.0.1 specification, which defines a secure infrastructure for post-issuance programmable smart cards. The OP specification defines a life cycle for OP compliant cards. State transitions between states of the life cycle involve well-defined sequences of operations. Cards, which have been issued to a Cardholder, are necessarily in a "SECURE" state. This means that a defined set of applications have been loaded onto the card plus a set of keys and a PIN through which the identities of the Cryptographic Officer and the Cardholder can be authenticated.

'*JCOP21id 32K'* is a single chip implementation of a cryptographic module. It is an ID-1 class smart card that adheres to the various ISO/IEC specifications for Integrated Circuit Chip (ICC) based identification cards. The 'cryptographic boundary" for the '*JCOP21id 32K'* card is the physical perimeter of the smart card. The module is comprised of the chip (ICC), the microelectronic connectors between the chip and contact pad, the contact faceplate, and plastic card. The module is constructed such as to provide the tamper resistance and the tamper evidence required by the FIPS 140-2 physical Level 3 validation.

The JCOP21id 32K chip is comprised of the following elements:

- Phillips P8WE5033 8- bit micro-controller with 32k EEPROM
- Systems software and cryptographic software installed in ROM as part of the chip manufacturing process. This hard mask is designated: JCOP21id 32K. This hard mask identification number can be retrieved from the card using a standard command. The Card Manager and PKCS15 applets are also masked in ROM

- Key and PIN storage in EEPROM as part of the card personalization operation.

2.2. Cryptographic Algorithms

The following FIPS approved cryptographic algorithms are available on the JCOP21id 32K.

- DES (ECB & CBC modes) (limited to legacy systems only)
- TDES (ECB, CBC modes, 128-bit key size)
- SHA-1
- DES MAC
- TDES MAC
- RSA PKCS #1 compliant Signature Generation
- AES (ECB, CBC modes; 128, 192, 256-bit key sizes)
- X9.31 Random Number Generation (compliant with Appendix 2.1 and 2.3) for RSA key generation

Additionally, the module provides the following non-FIPS approved algorithms

- AES MAC
- RSA Encryption/Decryption

3. Identities, Roles, Authentication, and Services

The *JCOP21id 32K* smart card provides identity-based authentication to prevent against unauthorized access, and the card assigns specific identities the privileges of a specific Cryptographic Officer Role or User Role.

3.1. Identities

The cryptographic subsystem of the *JCOP21id 32K* can authenticate the following identities.

- The **Card Issuer** represents the entity that ultimately owns the smart card itself. The Card Issuer authenticates by establishing a secure channel, thus proving knowledge of the card manager key set. The smart card always assigns this entity the privileges of the **Card Cryptographic Officer** role.
- The **First Card Holder Identity** represents the first of two entities that would possess the card on a day-to-day basis. The First Card Holder identity authenticates to the smart card by providing (within a PKCS#15 applet secure channel) his or her Personal Identification Number (PIN). The smart card always assigns the First Card Holder the privileges of the **Card Holder** role.
- The **Second Card Holder Identity** represents the second of two entities that would possess the card on a day-to-day basis. The Second Card Holder identity, like the first, authenticates to the smart card by providing his or her PIN (again, within a PKCS#15 applet secure channel). The smart card always assigns the Second Card Holder the privileges of the **Card Holder** role.
- Finally, the **Card Supervisor Identity** represents a third entity that would physically possess, although, only from time to time, as this entity can reset and change a Card Holder PIN value. This entity also authenticates to the smart card by providing (within a PKCS#15 applet secure channel) his or her PIN, and to allow the Card Supervisor to manage the Card Holders, the smart card always assigns the Card Supervisor entity the **Application (PKCS#15) Cryptographic Officer** role.

In addition to providing services to authenticated entities, the smart card also provides public or unauthenticated services that to any entity with physical access. The smart card provides all unauthenticated entities with the **Anybody** role.

3.2. Roles

As described above, the *JCOP21id 32K* smart card provides each entity with one of four roles. Additionally, each authenticated entity receives a specific role. The four roles provided by the smart card are

- $ROLE\_CO_{CARD}$. **The Card Cryptographic Officer** role has control of the services of the Card Manager, and thus controls the card's overall security configuration.
- $ROLE\_CO_{APP}$. The **Application (PKCS#15) Cryptographic Officer** role manages the security of the PKCS#15 applet. Most notably, this role can reset or change a Card Holder PIN value.
- $ROLE\_USER_{CARD}$. The **Card Holder** role can utilize the cryptographic services of the smart card (digital signatures, encryption, and decryption) and securely store data as well as cryptographic keys.
- $ROLE\_USER_{ANY}$. This role allows **Anybody** restricted access to the smart card's functionality. The Anybody role can solicit status information from the smart card.

Additionally, each of the four roles can be categorized as either belonging to a FIPS 140-2 Crypto-officer or User role. The Card Cryptographic Officer and the Application (PKCS#15) Cryptographic Officer roles

are considered to be FIPS 140-2 Crypto-Officer roles, while the Card Holder and Anybody roles are considered to be User roles.

3.2.1. User Roles

Anybody
Anybody is a generic role that the smart card provides those services that do not require authentication. This role does not grant access to any keys or CSPs contained in the module. Only basic commands required prior to authentication are accessible from this role. This includes the Select APDU to select an applet to authenticate to and the commands required to establish a Secure Channel. In addition as part of Card Manager applet, the Get Data and Get Status commands provide card status information.

Card Holder
The Card Holder represents an off-card entity using the services offered by the PKCS#15 applet. He is responsible for insuring the ownership of his card and for not communicating his PIN. The Card Holder is provided to an authenticated Card Holder entity after PIN verification.

3.2.2. Cryptographic Officer Roles

Application (PKCS#15) Cryptographic Officer
The PKCS#15 Cryptographic Operator represents an off-card entity managing the PKCS#15 applet. This management includes unblocking locked PINS, defining PKCS#15 resources and importing application keys. The external authentication requirement is configured per resource during personalization to the master PIN (PIN3).

Card Cryptographic Officer Role
The Card Cryptographic Officer has ultimate responsibility for managing the cards security configuration. This includes that of the individual security domains and the applications contained within. The Card Officer entity possess knowledge of the Card Manager Key Set, and has therefore access to the Card Cryptographic Officer Role and to the services offered by the Card Manager. In terms of key management, the officer is responsible for the secure creation of all security domains and the management of the security domain keys contained within.

3.3. Authentication

The module implements specific methods for authenticating the different users. The implementation consists of the binding of a user-based Access Control Rule to each service.

3.3.1. Card Holders and Card Supervisor Authentication

The First and Second Card Holder identities as well as the Card Supervisor identity authenticate to the smart card by providing their Personal Identification Number (PIN). The entity must provide the PIN within a secure channel in to access any Applet service protected with PIN. The APDU corresponding to the applet service must be sent before the card is removed or a reset order is send to the card. The verified PIN state persists until the applet is deselected or the card reset. In addition, an applet service that is protected with PIN Once requires that the entity authenticate before soliciting the applet command.

3.3.2. Card Issuer Authentication

The Card Issuer must prove the possession of the Card Manager key set in order to access any card manager service via secure messaging. Once the secure messaging channel is established, it will persist until the applet is deselected or the card reset.

3.4. Services

The applet services are invoked by external APDU commands sent to the card. The Access Control Rules (ACRs) are applied on the APDU commands.

3.4.1. Open Platform Services

The module provides the following services as specified in OP specification:
Delete, External Authentication, Initial Update, Install, Load, Put Data, Put Key, Get Status, and Set Status.

All these services are only available to the Cryptographic Officer owning the Card Manager keys. The Get Data service is available to any user. Please see the OP Specifications for descriptions of each of these functions.

Service	Description
DELETE	Used to delet an applet in EEPROM
EXTERNAL AUTHENTICATE	Authenticates the operator and establishes a secure channel.
GET DATA	Used to retireve a single data object
GET STATUS	Used to retrieve information about the card
INITIALIZE UPDATE	Used to open a secure channel with the card
INSTALL (INITIALIZE)	Use to initiate or perfrom the various steps required for Card Content management
LOAD	Used to load a load file (e.g. an applet)
PUT DATA	Used to transfer data to an application during command processing
PUT KEY	Used to replace the keys of the Card Manager
SELECT	Used to select an application (applet)
SET STATUS	Used to modify the life cycle status or the application (applet) life cycle state

The following table shows which OP services are accessible and which services allow read (R) or write (W) access to the Card Manager key set.

	ANYBODY	CARD CO	KEYSET ACCESS
DELETE		X	
EXTERNAL AUTHENTICATE		X	R
GET DATA	X		
GET STATUS		X	
INITIALIZE UPDATE	X		R
INSTALL (INITIALIZE)		X	
LOAD		X	R
PUT DATA		X	
PUT KEY		X	W
SELECT	X		
SET STATUS		X	

3.4.2. PKCS#15 Applet Services

The PKCS#15 applet provides a variety of services. The following table lists the different APDUs / Services that are provided by the PKCS#15 applet:

Service	Description

CHANGE REFERENCE DATA	Change the current reference data (e.g. PIN)
CREATE FILE	Creates a new file in the PKCS#15 file store
DELETE FILE	Removes a file from the PKCS#15 file store
ERASE BINARY	Erase the contents of a transparent (binary) file
GET CHALLENGE	Requests that the Card supply a challenge
MANAGE SECURITY ENVIRONMENT: **RESTORE**	Restore a predefined (or empty) security environment.
MANAGE SECURITY ENVIRONMENT: **SET**	Set the security environment (algorithms, keys) that shall be used in the following PERFORM SECURITY OPERATION commands.
MUTUAL AUTHENTICATE	Allows mutual authentication (in conjunction with GET CHALLENGE) between the host and the card and results in a secure channel.
PERFORM SECURITY OP: **COMPUTE DIGITAL SIGNATURE**	Compute a digital signature with a private key. The algorithm and key are specified with the Manage Security Environement command.
PERFORM SECURITY OP: **CRYPTO CHECKSUM**	Calculate. The algorithm and key are specified with the MSE command.
PERFORM SECURITY OP: **DECIPHER**	Decrypt data with a private key. The algorithm and key are specified with the MSE command.
PERFORM SECURITY OP: **ENCIPHER**	Encrypt data with a public key. The algorithm and key are specified with the MSE command.
PERFORM SECURITY OP: **HASH**	Create a cryptographic hash of the given data. (SHA-1 only)The algorithm and key are specified with the MSE command.
PERFORM SECURITY OP: **GENERATE KEY PAIR**	Creates a new RSA key pair on the card.
PUT DATA	Sets the Read/Modify authentication objects to NEVER for the given file.
READ BINARY	Read binary data from a transparent (binary) file
RESET RETRY COUNTER	Unlock locked reference data (e.g. PIN)
SELECT FILE	Select a file from the card's file system
UPDATE BINARY	Update the contents of a transparent (binary) file
VERIFY	Verify reference data presented by user (e.g. PIN) with the reference data stored inside the card. The current verification status can be also queried with this command.

Service List

The following table shows which PKCS15 applet services are accessible in each role supported by the PKCS15 applet.

	CHANNEL SECURITY REQUIRED		**USER**		**OFFICER**
	AU	**SM**	**ANYBODY**	**PIN1/PIN2**	**PIN3**
CHANGE REFERENCE	-	Yes	No	No	Yes

DATA					
CREATE FILE[1] Default *Create* Auth.: PIN1,PIN2, Default *Create* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
CREATE FILE Default *Create* Auth.: AUTH	Yes	Yes[2]	No	Yes	Yes
CREATE FILE Default *Create* Auth.: SM	-	Yes	No	Yes	Yes
DELETE FILE File *Delete* Auth.: PIN1,PIN2, File *Delete* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
DELETE FILE File *Read* Auth.: AUTH	Yes	Yes	No	Yes	Yes
DELETE FILE File *Read* Auth.: SM	-	Yes	No	Yes	Yes
ERASE BINARY (Non/Pub Key) File *Modify* Auth.: PIN1,PIN2, File *Modify* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
ERASE BINARY (Non/Pub Key) File *Modify* Auth.: SM	-	Yes	No	Yes	Yes
ERASE BINARY (Non/Pub Key) File *Modify* Auth.: AUTH	Yes	Yes	No	Yes	Yes
ERASE BINARY (Priv/Sym Key)	-	Yes	No	No	No
GENERATE PUBLIC KEY PAIR File(1&2) *Modify* Auth.: PIN1,PIN2, File(1&2) *Modify* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
MANAGE SECURITY ENVIRONMENT: (RESTORE)	-	-	Yes	Yes	Yes
MANAGE SECURITY ENVIRONMENT: (SET)	-	-	Yes	Yes	Yes
PERFORM SECURITY OP ENCIPHER File *Encipher* Auth.: PIN1,PIN2, File *Encipher* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
PERFORM SECURITY OP: COMPUTE DIGITAL SIGNATURE File *Sign* Auth.: PIN1,PIN2, File *Sign* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes

[1] Default Create Authority and initialization parameter

[2] Authorize (AU) is a subset of Secure Messaging (SM)

PERFORM SECURITY OP: DECIPHER File *Decipher* Auth.: NO PIN File *Decipher* Auth.: PIN1,PIN2, File *Decipher* Auth.: PIN3	-	-	 No No	 Yes Yes No	 Yes No Yes
PERFORM SECURITY OP: HASH	-	-	Yes	Yes	Yes
PERFORM SECURITY OP: CRYPTO CHECKSUM	-	-	No	Yes	Yes
PUT DATA File *Modify* Auth.: PIN1, PIN2 File *Modify* Auth.: PIN3 File *Modify* Auth.: AUTH File *Modify* Auth.: SM	- Yes	- Yes Yes	 No No No No	 Yes No Yes Yes	 No Yes Yes Yes
READ BINARY (Non/Pub Key) File *Read* Auth.: SM	-	Yes	Yes	Yes	Yes
READ BINARY (Priv Key)	-	-	No		
RESET RETRY COUNTER	-	Yes	No	No	Yes
SELECT FILE	-	-	Yes	Yes	Yes
UPDATE BINARY (Non/Pub key) File *Modify* Auth.: PIN1, PIN2 File *Modify* Auth.: PIN3	-	-	 No No	 Yes No	 No Yes
UPDATE BINARY (Non/Pub Key) File *Modify* Auth.: AUTH	Yes	Yes	Yes	Yes	Yes
UPDATE BINARY (Non/Pub Key) File *Modify* Auth.: SM	-	Yes	Yes	Yes	Yes
UPDATE BINARY (Priv Key)	-	Yes	No		
VERIFY		Yes	Secure Messaging Enforced		

Notes to above table

AU: Authentication enforced
SM: Secure Messaging Enforced

The default *Create* authority is an applet initialization parameter
The *Read*, *Modify*, *Sign*, *Encipher*, *Decipher*, *Delete* authorities are parameters during file creation
The **Put Data** service sets the given file's modify and read authority to NEVER
The **Generate Public Key Pair** service sets the private key file's *Read* and *Modify* authorities to NEVER.
Secure Messaging is a superset of Authorize (AU) thus where the channel security requirement is a minimum of AU, SM can be used.

4. Security Rules and Secure Operation

In order to operate the *JCOP21id 32K* securely, the operator should be aware of the security rules enforced by the module and should adhere to the physical security rules and secure operation rules required.

4.1. Security Rules

4.1.1. JCOP21id 32K Security Rules

The following are security rules that result from the security requirements of FIPS 140-2 include the following rules:

Applet Environment

- All post issuance applets must be installed within a FIPS 140-2 validated smart card.
- All post issuance applets must be FIPS140-2 validated
- The Card Holder must take the necessary measures to insure that the terminal and/or the Card Acceptance Device are controlled by a valid role: Card Holder, Application Operator or Cryptographic Officer.

Content Management

- The management of the life cycle of the applets – load, install, delete, personalize keys, shall follow the Open Platform standard.
- Applets management and key management APDU commands, also called services, (such as download, install, delete, put key) are protected by the Card Secure Channel encryption, authentication and signing. They have their origin authenticated and their integrity verified. In particular this protects the applet byte code against tampering when loaded at post issuance.

Authentication

- The applets shall provide the following distinct operator roles:
  - The User or Card Holder role
  - The Application Cryptographic Officer
- In those instances where all of the cryptographic officer roles are assigned to one entity, a common key set may be defined.
- The applets shall provide identity-based authentication.
- Cryptographic services are restricted to authenticated roles.
- When authentication of the role cannot be performed because the related key or password or key attributes are missing, the corresponding service will not be made available.
- The results of authentication must be set in transient memory and therefore cleared when the module is powered down.

Key management

- The applet must force the encrypted transport of 3DES keys to the card.
- RSA private keys may not be imported on the card
- The Card Manager Keys may only be divulged to the Card Cryptographic Officer.
- The Application security keys may only be divulged to the Application Cryptographic Officer, and the Card Cryptographic Officer.
- The application specific keys may only be divulged to the card holder, the Application Cryptographic Officer, and the Card Cryptographic Officer.
- The size of files holding symmetric (DES/TDES/AES) keys must equal the size of the key. Consequently, the size of such key files can be in the range of 8-32 bytes.
- Read, update and erase operations on secret or private key files are only allowed if secure messaging is used. To enforce this it is not possible to access files smaller than 33 bytes that allow any of the crypto operations (sign, encrypt, decrypt) without secure messaging. The same is true for larger files that allow sign or decrypt operations since these files might hold RSA private

keys. Files greater than 32 bytes that allow the encryption operation only, however, can be updated without secure messaging since they can only be used for public key operations.

PIN management

- The password or PIN that is used by the applet to authenticate the Card Holder must not be divulged to other parties than the Card Holder.
- The PKCS#15 applet (also known as the ID applet) must be configured by the cryptographic officer so that:
- After M consecutive unsuccessful PIN code validation attempts, the Card Holder services must be disabled. (eg. The PIN is blocked). In FIPS mode M = 3
- After N consecutive unsuccessful PIN unblocking attempts using OP authentication with incorrect key or parameters, the card Holder services are permanently disabled (eg. The PIN is locked). In FIPS mode N = 10
- The PKCS#15 PINs have a length of 16 bytes

Card Issuer's Security Responsibilities

The Card Issuer is responsible for:

- Generating and loading the Issuer Security Domain keys,
- Enforcing standards and policies for Application Providers governing all aspects of Applications to be provided to the Card Issuer or operated on the Card Issuer's cards,
- Working with Application Providers to create and initialize Security Domains other than the Issuer Security Domain,
- Determining policy with regards to card and application Life Cycle management, velocity checking levels, Application privileges, and other security parameters,
- Managing the application code loading and installing both on a pre-issuance and post-issuance basis, and
- Cryptographically authorizing load, install, and extradition to be performed by Application Providers.

Application Provider's Security Responsibilities

The Application Provider (a subordinate delegate of the Card Manager) is responsible for:

- Generating the keys for its own Security Domains or obtaining Security Domain keys from a trusted third party,
- Working with the Card Issuer to load generated keys into the Application Provider's Security Domain,
- Providing applications that meet the Card Issuer's security standards and policies,
- Providing application code signatures according to the Application Provider's security policy,
- Obtaining pre-authorization for load, install, and extradition from the Card Issuer, and
- Returning Load, Install, Delete, and Extradition Receipts, according to the Card Issuer's policy.

Card Holder's Security Responsibilities

- The Card Holder shall keep the card within his/her possession at all times
- The Card Holder shall not divulge the PIN value and PKCS15 Security keys to anyone
- The Card Holder shall only use FIPS approved algorithms in the FIPS mode of operation

4.1.2. Physical Security Rules

The physical security of the *JCOP21id 32K* module is designed to meet FIPS 140-2 level 3 requirements. The module manufacturing process consists of wire-bonding the P8WE5033 ICC to a printed circuit plate providing the ISO contacts. The plate and ICC are subsequently sealed in epoxy coating to become Modules. These Modules are subsequently sent to Card Manufacturers where they are embedded in a plastic card body to produce the JCOP21id 32K smart card. The epoxy coating surrounding the module cannot be penetrated without leaving evidence of the attack and attempts to mechanically extract the chip from the module will result in damage to the chip. Further, the packaging itself is resistant to penetration

and attempts to mechanically extract modules from the smart card body will result in signs of tampering such as scratches and deformation.

From the time of its manufacture, the JCOP21id 32K is in the possession of the Cryptographic Officer until it is ultimately issued to the User. From that point, the card is in the physical possession of the User. To compromise the cryptographic information contained within the module requires physical access to the card. To eavesdrop on the normal activities of the module, while it is still in possession of either one of the Cryptographic Officers or of the User, will be demonstrated to be difficult or impossible due to the protocols and security mechanisms protecting access to the module's information and services. To eavesdrop on the module through extraordinary means requires physical possession of the card. In this event, either the Cryptographic Officer or the User detects the absence of the card and triggers the disabling of the capabilities of the card within a larger systems context. If the module is attacked through physical means, the attack will be evident due to the disturbance of the packaging of the card and module.

The P8WE5033 ICC also provides the JCOP21id 32K with various hardware security mechanisms such as a clock input filter to for protection against spikes and offers voltage, frequency and temperature sensors, each with a high and low safeguard. These are detailed in Attack Mitigation.

4.2. Secure Operation Initialization Rules

The JCOP21id 32K smart card supports only FIPS-Approved algorithms. However, the module is in the FIPS mode when the PKCS15 applet is instantiated and personalized with certain parameters described below.

4.2.1. Applet Installation

As part of the PKCS15 applet installation, certain install parameters are passed that allocate resources for the default file system layout. These parameters also define certain security settings. The Card CO can specify the reset retry counter limit for each user and the security attributes that protect the key files used for authentication. These settings also determine whether private keys are importable on the card or not. The table below explains each install parameter in detail.

Byte	Value[HEX]	Remarks
1	C9	Application specific install parameters tag (see [9])
2	1A	Length
3	XX	RFU
4	XX	RFU
5	XX	RFU
6	XX	Overall file system space(high byte)
7	XX	Overall file system space(low byte)
8	XX	Size EF(PuKDF) (high byte)
9	XX	Size EF(PuKDF) (low byte)
10	XX	Size EF(PrKDF) (high byte)
11	XX	Size EF(PrKDF) (low byte)
12	XX	Size EF(CDF) (high byte)
13	XX	Size EF(CDF) (low byte)
14	XX	Size EF(AODF) (high byte)
15	XX	Size EF(AODF) (low byte)
16	XX	Size EF(DODF) (high byte)
17	XX	Size EF(DODF) (low byte)
18	XX	Size EF(SKDF) (high byte)
19	XX	Size EF(SKDF) (low byte)
20	XX	Size EF(ODF) (high byte)
21	XX	Size EF(ODF) (low byte)

22	XX	Size EF(TokenInfo) (high byte)
23	XX	Size EF(TokenInfo) (low byte)
24	01-7F	Retry counter limit for PIN 1
25	01-7F	Retry counter limit for PIN 2
26	01-7F	Retry counter limit for PIN 3
27	XX	Zero means RSA private keys are importable, else otherwise
28	01-05	Defines the authentication object, which protects the CREATE FILE Command Values: 1 - PIN 1 2 - PIN 2 3 - PIN 3 4 - AUTH 5 - SM

In order to be FIPS mode of operation the PKCS15 applet install parameters must have the following values

C9	Tag
1a	Length
000000	RFU
1c00	7K file system
00c8	200 byte puKDF
00c8	200 byte prKDF
00c8	200 byte CDF
0096	150 byte AODF
00c8	200 byte DODF
0032	50 byte SKDF
0064	100 byte ODF
0078	120 byte TokenInfo
03030a	Retry counter limits for PIN 1-3 (PIN1=3 PIN2=3 PIN3=10)
01	Imported keys disallowed
xx	Defining the authorization object which protects the create file command. This can be set to any value.

Once the PKCS15 applet is installed on the JCOP21id 32K smart card it needs to be personalized by the Card CO. Personalization of the applet involves defining initial PKCS15 Security keys and PINs of the applet in a secure manner. A Card CO needs to authenticate himself to the Card Manager applet in order to perform PKCS15 applet personalization.

The personalization of the PKCS#15 application is based on certain Open Platform functionality. For detailed information about Open Platform application life cycles, Open Platform secure messaging etc. see [9]. Upon installation of the PKCS#15 application, the application life cycle state is “Selectable”. In this state it only accepts a special set of commands, which allow the applications initial personalization via Open Platform secure messaging using Card Manager keys. After successful personalization, the life cycle of the application is transitioned to “PERSONALIZED”. See the description of the command below for details of the personalization process.

4.2.2. Personalization

Open Platform INITIALIZE UPDATE Command

This command initiates the personalization process. Together with the Open Platform EXTERNAL AUTHENTICATE COMMAND it carries out mutual authentication between the card and the host application and sets up a secure channel. This is based on symmetric keys of the Open Platform Card Manager. After successful processing of the command the card is authenticated and it expects the Open Platform EXTERNAL AUTHENTICATE COMMAND.

Command APDU:

Code	Value [HEX]	Remarks
CLA	80	Proprietary
INS	50	INITIALIZE UPDATE command
P1	00 or 01 to 7F	Key set version
P2	00	Key index
LC	08	Length of data
Data	XX	Host challenge
LE	00	

Response APDU:

As defined in [9].

Open Platform EXTERNAL AUTHENTICATE Command

This command authenticates the host and completes the setting up of the secure channel. A previous and successful INITIALIZE UPDATE COMMAND is necessary prior to processing this command. The security level “Encryption and MAC” is mandatory for token personalization.

Command APDU:

Code	Value [HEX]	Remarks
CLA	84	Proprietary with secure messaging
INS	82	EXTERNAL AUTHENTICATE command
P1	03	Security level
P2	00	Parameter
LC	10	Length of data
Data	XX	Host cryptogram and MAC
LE	-	Not present

Response APDU:

Empty (status word 9000_{HEX} only).

INITIALIZE TOKEN Command

Once the secure channel is set up the token can be personalized. Using this command the initial secure messaging keys (for ISO secure messaging) and the initial PIN values are set on the token. Upon successful processing of this command the life cycle state of the application is set to “PERSONALIZED”.

Command APDU:

Code	Value [HEX]	Remarks
CLA	84	Proprietary with secure messaging
INS	F0	EXTERNAL AUTHENTICATE command

P1	00	
P2	00	
LC	51	Length of data
Data	XX	MACed and encrypted (Open Platform): 16 byte - MAC key 16 byte - ENC key 16 byte – PIN 1 16 byte – PIN 2 16 byte – PIN 3 1 byte – RFU
LE	-	Not present

Response APDU:

Empty (status word 9000_{HEX} only).

Once the module is in FIPS mode of operation, the FIPS mode indicator can be obtained by calling the Card Manager get Status command. The Card Manager state is set to **Secured** and the PKCS15 applet state is set to **Personalized** in the FIPS mode of operation. The module will be in the FIPS mode if the installation and personalization procedures have been followed correctly.

5. Definition of SRDIs Modes of Access

This section specifies the *JCOP21id 32K* smart card's Security Relevant Data Items as well as the access control policy enforced by the smart card.

5.1. Cryptographic Keys, CSPs, and SRDIs

List of card-managed security related data items

- **Authentication Data**: These are the PIN1, PIN2, and PIN3 used by the different PKCS#15 identities to authenticate.
- **External Authentication Keys**: These are 16-byte 3DES keys that enable the authentication of the service roles.
- **RSA private keys**: are managed (generated, unwrapped) from the PKCS#15 applet using the java card cryptographic services. These keys are used to sign data.
- **RSA public keys**: Public keys are generated on card from the RSA key pair generation, and exported off card.
- **X.509 Certificates**: The certificates corresponding to the private keys present in the card are managed by the applets.
- **Personal Identification Numbers or passwords (**PIN): PINs and PIN attributes are managed from within the application (PKCS#15) applet by the Application Cryptographic Officer.
- **Open Platform Key Sets:** consist of 112-bit TDES keys and are managed by the card manager or security domain. These keys enable the authentication of the Card Cryptographic Officer These key sets also allow the encryption of keys which are input into the JCOP21id 32K via the Put Key command.

5.2. Software Design

The basic systems software of JCOP21id 32K is secure from modification due to the fact that it is stored in ROM. The systems software is written primarily in the C and Java programming languages that allows for extensive review to confirm security. Some speed critical functions have been implemented in assembler such as the RNG and cryptographic algorithms. Such speed optimization is essential in resource constrained environment like smart cards in order to meet performance criteria.

The JCOP21id 32K implements the JavaCard™ 2.1.1 standard. This includes an on-card Java Card Virtual Machine and firewall. Applets are secure from each other due to the fact that each runs in a "Java sandbox" where the firewall protects applet objects from illegal access by another applet Applets. The Java Card language does not contain any constructs that allow cross-sandbox communication directly; any such communication is through controlled systems software mechanisms which allow the implementation of strict security measures.

The software security of the '*JCOP21id 32K*' card is strictly controlled by the OP Card Manager application. The OP specification defines extensive access control rules that serve to counter any threat that users might *attempt to delete applications without authorization* and that users may *attempt to do things, which are outside of their intended authorization*. Some of these rules are discretionary in nature, the discretion being in the hands of the card issuers to approve access rights to application providers and applications. Other rules are mandatory in nature and are imposed by the OP specification in the form of state transitions within a finite state machine framework. For more information see the separate document BlueZ: JCOP21id Technical Description.

The post issuance download of applets onto the JCOP21id 32K module is according to the Open Platform 2.0.1 specifications for Secure Channel. This specification requires that a Card Manager secure channel be established in order to upload applets on the card. This does not impact the PKCS#15 applet since it is already is masked into ROM. Please note that in order to maintain FIPS compliance only the following applets may be loaded onto the card post issuance: **none** (currently no additional applets have been FIPS validated).

5.3. File System

The PKCS15 application provides an emulation of a PKCS#15 compliant ISO file system layout in combination with the ISO 7816 Parts 4-6 commands. Each file has a set of Security Attributes associated with it. These access conditions determine whether the file system commands Read Binary, Update Binary, Erase Binary and Delete File can be performed on the file within the current authorization level. These nibbles also determine whether cryptographic operations are allowed using the key contained in the file. These attributes are set during file creation (by the Create File command). The file creation is bound to one of the PINs (as defined during token installation) and thus can only be performed if this PIN is verified.

The security attributes of a file are encoded in three bytes whereas each nibble represents the access conditions for a certain type of operation as can bee seen below:

Byte 1		Byte 2		Byte 3	
READ	MODIFY	SIGN	ENCIPHER	DECIPHER	DELETE

The access conditions encoded in each of the nibbles are defined as follows:

Bit 4	Bit 3	Bit 2	Bit 1	Condition
-	0	0	0	ALWAYS
-	0	0	1	NEVER
X	0	1	0	PIN 1
X	0	1	1	PIN 2
X	1	0	0	PIN 3
-	1	0	1	AUTH
-	1	1	0	SM
-	1	1	1	RFU

Bit 4 is the ***one time*** bit. If this bit is set the associated PIN will be invalidated after each operation for which the PIN needs to be verified. This allows, for instance, requiring PIN validation before each private key usage for signing. It is to be noted that the access conditions AUTH and SM both relate to the secure channel (established via a MUTUAL AUTHENTICATE command). AUTH indicates a secure channel with no encryption (mutual authentication only) and SM indicates authentication with subsequent secure messaging. Consequently, if secure messaging is done (SM) the AUTH condition is satisfied as well but not vice versa.

6. Key Management

The JCOP21id 32K card manages the following keys:

Initialization Keys

These keys are used during the manufacturing process for the module's pre-initialization and secure distribution. All keys are double length TDES keys

1. K_{init_cm} used only for the initial Card Manager key-set loading,
2. K_{init_fab} fabrication key used only for personalization of the FABKEY area
3. $K_{init\_transport}$ transport key for root applet access.

.

These keys are not accessible in the post-issuance phase once the card has been issued to the user, thus these keys are not accessible or used while the card is operating in its FIPS mode.

Card Manager Keys

The JCOP21id 32K implements the Open Platform 2.0.1 specification which specifies that a Security Domain, including the Issuer Security Domain shall have at least one key set containing 3 keys to be used in the initiation and use of a Secure Channel. These keys are all 112-bit TDES keys and are the following:

a) The Secure Channel Encryption/Authentication key (K_{cm_auth}) which is used to set up an encryption session key for mutual authentication and to encrypt APDU command data.

b) The Secure Channel MAC Key (K_{cm_mac}), which is used to generate a MAC session key for TDES MACing APDU data

c) The Data Encryption Key (K_{cm_ek}) for decrypting sensitive data, e.g. secret or private keys. This key is a 112-bit TDES key and is used as a static key.

PKCS#15 security keys

The PKCS#15 Applet contains 2 static 112-bit TDES keys for secure channel establishment. These keys are defined during applet initialization. These keys belong to the PKCS#15 application for generating a secure channel and are not used for authentication.

a) The Secure Channel Encryption/Authentication key (K_{pkcs_auth}) which is used to set up an encryption session key for mutual authentication and to encrypt APDU command data.

b) The Secure Channel MAC Key (K_{pkcs_mac}), which is used to generate a MAC session key for MACing APDU data

PKCS#15 application keys

Applets such as the PKCS#15 applet may use various keys and key types with the cryptographic services that the module offers. The number and type of keys will vary depending on the external application, but will be constrained to the FIPS-approved algorithms provided by the card (DES, TDES, AES, and RSA [PKCS#1]).

4. K_{pk_app} application specific keys stored within in the PKCS#15 file system

7. Self-tests

The *JCOP21id 32K* performs a number of self-tests to ensure that cryptographic functions operate correctly.

The power-up self-tests include:

- An EEPROM integrity check of all uploaded packages
- An EEPROM integrity check of the native jump table
- An EEPROM integrity check of all code patches
- Cryptographic Algorithm tests for:
  - Pseudo Random Number Generator
  - DES
  - 3DES
  - SHA-1 Hashing
  - RSA sign/verify
  - RSA encrypt / decrypt
  - AES

If any of the above tests fail, the card will enter an error state in which further APDU's are not processed. No data of any type is transmitted to the CAD. A card-reset signal (RST) is required for the card to re-activate, whereby the same initial self tests will be executed.

The following conditional self-tests are also performed by the module:

- Pairwise Consistency test on RSA key generation
- Continuous Random Number Generator test when the FIPS PRNG is invoked
- Statistical RNG tests (monbit and poker tests) can be called by the operator on 20,000 bits of random data collected from the PRNG

8. Attack Mitigation

8.1. Power Analysis Attacks (Simple and Differential)

Description: DPA attacks exploit characteristic behaviors of transistor logic gates and software running on today's smart cards. The attacks are performed by monitoring the electrical activity of a device, then using advanced statistical methods to determine secret information (such as secret keys and user PINs) in the device.

Mitigation: The JCOP21id 32K uses the secure hardware cryptographic processor. Furthermore, the hardware introduces random wait states on EEPROM write operations, power up and down resets. DES keys are read in random order from the EEPROM and a key obfuscation option is available using a secret ROM value. The RSA algorithm implements a block wise exponent function hash instead of bit-wise evaluation.

8.2. Timing Analysis Attacks

Description: Timing attacks are non-invasive attacks that rely on the variation in computation time required for the CPU on a smartcard to perform its secret calculation.

Mitigation: The JCOP21id 32K RSA key generation algorithms ensure that each response takes the same length of time by factoring RSA keys. The hardware also provides self timed EEPROM programming which is hardware sequencer controlled. A further countermeasure is the on-chip high voltage generation.

8.3. Fault Induction

Description: One such attack is the Bellcore attack against tamper-resistant chips such as smart cards. The attack is based on the (theoretical) possibility of flipping some bits (at some random position) of the secret key, stored in RAM or E2PROM, before or during the computations done by the chip. Another attack is to induce some decoding error during the execution of one instruction (Anderson and Kuhn).

Mitigation: The hardware use a clock input filter to for protection against spikes and offers voltage, frequency and temperature sensors, each with a high and low safeguard. The hardware also provides memory protection for RAM, EEPROM and ROM. Finally, the hardware runs an on-chip self test with signature technique.

8.4. Other

PKCS#15 applet resets its PIN verified state when a secure channel is closed and when the applet is deselected. Ten successive failed attempts at building a secure channel will block the PKCS#15 applet.