JCOP21id Technical Brief

Overview: *This document contains a simple overview about the technical capabilities of the ID market family member of the JCOP card OS family. Requests for further information may be directed at javacard@zurich.ibm.com.*

1. Basic specifications

JCOP is an IBM BlueZ implementation of the basic specifications [1] and [2] including refinements from Visa International set in the Visa OpenPlatform Card Implementation Requirements (http://www.visa.com/nt/suppliers/vendor). All necessary clarifications from ISO7816 and EMV 2000 are also incorporated into the implementation where required by [1] and [2].

JCOP21id is the ID industry member of this family. It adds FIPS 140-2 level 3 compliance [3] as well as support for multiple SecurityDomains and DAP verification to the ones listed for JCOP20.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default][1]

ISO7816 T=0 direct convention[1]

ISO7816 T=1 inverse convention[1]

ISO7816 T=0 inverse convention[1]

2.2 Supported speeds

At the default clock rate of 3.57 MHz, the following communication speeds can be attained:

9600 bit/sec [default]

19200 bit/sec

38400 bit/sec

57600 bit/sec

115200 bit/sec

[1] The contact protocols of JCOP can be configured to support the clock-stop feature of certain terminals to save power consumption (typically done in mobile phones).

3 Memory availability for applications

3.1 EEPROM

3.1.1 Persistent Java heap

Used for allocating persistent objects, applets, and storage of post-issuance loaded applet code (aka packages).

Size: 14/30kB* (with FIPS-compliance applet [4] in ROM).

3.1.2 Transaction buffer

Used to save transactional written data (e.g. all persistent byte and short stores), as well as persistent parameters to Util.arrayCopy; see [1].

Size: 512 bytes

3.2 RAM

3.2.1 Transient Java heap

Used for allocating transient objects and arrays of type CLEAR_ON_RESET and CLEAR_ON_DESELECT.

Size: 512 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communication data.

Size: 261 bytes

3.2.3 Java stack

Used to hold call parameters, local variables, and stack frames of the VM.

Size: 200 bytes

3.3 ROM

8/40kB* free for further applications beyond the on-board FIPS-compliance applet [4].

* Depending on hardware configurations (see section 5)

4 Supported optional features

Certain features listed in the JavaCard and OpenPlatform standards are not defined to be mandatory. Those which have been implemented in JCOP are listed below.

4.1 JavaCard

4.1.1 Garbage Collection

Fully implemented: Deleted objects, applets, and packages are fully reclaimed and the space can be used for other purposes after deletion.

4.1.2 Standard Cryptographic Algorithms

JCOP21id has the ability to generate RSA keys on the card. The following JavaCard API constants (see [1]) are implemented by this version of JCOP:

Ciphers:

ALG_DES_CBC_NOPAD

ALG_DES_CBC_ISO9797_M1

ALG_DES_CBC_ISO9797_M2

ALG_DES_ECB_NOPAD

ALG_DES_ECB_ISO9797_M1

ALG_DES_ECB_ISO9797_M2

ALG_RSA_PKCS1

ALG_RSA_NOPAD

Signatures:

ALG_DES_MAC8_NOPAD

ALG_DES_MAC8_ISO9797_M1

ALG_DES_MAC8_ISO9797_M2

ALG_RSA_SHA_ISO9796

ALG_RSA_SHA_PKCS1

ALG_RSA_MD5_PKCS1

MessageDigest:

ALG_SHA

ALG_MD5

RandomData:

ALG_SECURE_RANDOM

KeyTypes:

LENGTH_DES

LENGTH_DES3_2KEY

LENGTH_RSA_2048[1]

LENGTH_RSA_1024[1]

LENGTH_RSA_768 [1]

LENGTH_RSA_512[1]

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

TYPE_RSA_PRIVATE[2]

TYPE_RSA_CRT_PRIVATE

KeyPair:

ALG_RSA_CRT

4.2 OpenPlatform

4.2.1 Global PIN

Fully implemented: All described APDU and API interfaces for this feature are present.

4.2.2 Security Domains/DAP

JCOP21id allows the installation of multiple Security Domains as well as Mandated DAP verification.

4.3 FIPS 140-2

The JCOP21id cryptographic algorithms have been certified to their respective FIPS standards as part of the JCOP21id FIPS 140-2 validation. The complete system with a PKCS#15 applet is scheduled to receive FIPS140-2 level 3 certification by June 2003. (rf. http://csrc.ncsl.nist.gov/cryptval).

[1] All multiples of 32 (bit) are supported as valid RSA key lengths. Thus, key length values such as 736 (bits) can be passed as parameters to the respective functions.

[2] Private Keys must be loaded with key material. On-card key generation is only supported for RSA keys in CRT format

5 Supported Hardware

The same JCOP mask can be combined with any of the following platforms and automatically make use of their different resources capabilities, i.e., create a larger persistent Java heap, and make more ROM memory available for ROMed applets*.

5.1 Philips P8WE5009

96 kB ROM → 40kB free for ROM'd applets in Custom Mask Process

8 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.2 Philips P8WE5017

64 kB ROM → 8kB free for ROM'd applets in Custom Mask Process

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.3 Philips P8WE5033

96 kB ROM → 40kB free for ROM'd applets in Custom Mask Process

32 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

* Custom applets may be submitted to Philips for inclusion into a ROM mask. The maximum package size is 16kB. The Custom Mask process is only available starting with JCOP10v2.

6 Performance figures

In the absence of standard performance tests, typical applet's operations are timed. The protocol used is T=1 at 9600 bit/sec. The reader clocks the chip at 4 MHz. The applets are the Visa approved versions after having been initialized and populated with keys as required for Visa testing. To avoid measuring communications overhead, timing is measured between the last APDU byte sent to the reader and the first byte returned from the card.

Operation	ETUs	msec
SELECT CardManager[1]	141	13.1
INIT UPDATE CardManager[1]	431	40.1
EXTERNAL AUTH CardManager[1]	299	27.8
Install VisaCash[2]	5617	522.4
SELECT VisaCash[1]	156	14.5
Initialize LOAD for VisaCash[1]	821	76.4
Perform LOAD for VisaCash[1]	1574	146.4
Initialize PURCHASE for VisaCash[1]	229	21.3
Perform PURCHASE for VisaCash[1]	1668	155.1
ReadBalance from VisaCash[1]	101	9.4
SELECT VSDC[1]	185	17.2
GenerateAC from VSDC[1]	2193	203.9

The PK operations are largely dominated by the hardware speed of the Philips FameX PK coprocessor. Note that on-card key generation is a random-based process; thus the figure given is only an average value. Values are measured at low-level; depending on Java programming skills, application-level code can add some time to the values depicted here.

Operation	Msec
1024 bit CRT public key operation (F4)	33
1024 bit CRT private key operation	417
Generate 1024bit CRT key	~3800
Generate 2048bit CRT key	~62000

[1] Second time command is executed (to eliminate potential applet setup effects)

[2] On 'clean' card (to eliminate potential EEPROM clearing effects)

A Revision History

1.0	Initial document
1.1	Specification amendments (4.1.2, 5.1), Free ROM applet size clarified (5)
1.2	Added comment on clock-stop feature (2.1)

B References

[1] Sun Microsystems: JavaCard 2.1.1 http://java.sun.com/products/javacard

[2] Global Platform Consortium: OpenPlatform 2.0.1' http://www.globalplatform.org/

[3] FIPS PUB 140-2: Security Requirements For Cryptographic Modules, May 2001, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[4] International Business Machines Corp.: PKCS#15 Applet User Guide, rev. 1.3, September 2002.