JCOP –
The IBM GlobalPlatform JavaCard™ implementation

Highlights

- ***Fully standards compliant***
  - ***JavaCard 2.1.1***
  - ***GlobalPlatform 2.0.1'***
  - ***EMV & ISO7816***
  - ***ISO14443***
  - ***3GPP 03.19 & 11.14***
- ***Uses least expensive hardware:***
  - ***8 bit controller***
  - ***48k of ROM***
  - ***Custom Mask Process***
- ***Industry-leading performance in benchmarking and applications***
- ***Full feature set:***
  - ***RSA (2048 bits) with on-card key generation***
  - ***contact/contact-less interface***
- ***Guaranteed interoperability***
- ***Cross-industry technology***
  - ***Financial/EMV market***
  - ***Telecom/SIM market***
  - ***ID/health market***

JCOP is the open IBM smart card operating system based on independent, third-party specifications, e.g., by Sun Microsystems, the GlobalPlatform consortium, the International Organization for Standardization (ISO), EMV, and others. JCOP is available for licensing across the smart card value chain and across industries. JCOP guarantees interoperability of applications in an unprecedented manner.

Use of standards
The JavaCard, GlobalPlatform and ISO industry standards together ensure application interoperability for card issuers as well as solutions developers. By adhering not just to the standards themselves, but also to their spirit as evidenced in numerous heritage applications, JCOP ensures 100% interoperability with third-party applets as well as all existing smart card infrastructures. This way JCOP makes true on the promise of multi-sourcing any component in smart card solutions. Even in existing infrastructures, JCOP equipped with proper applications can substitute any existing proprietary smart card infrastructure.

Least expensive hardware
From the start, JCOP has been developed for small, standard 8 bit smart card controllers.

By supporting a wide range of different versions of chip hardware in an easy to configure manner, JCOP is available for low-cost as well as high-end microcontroller configurations.

High Performance
On cross-platform tests, JCOP consistently beats competitive JavaCard implementations. This includes other implementations running on 32bit platforms. Even most proprietary cards are beat on the only measure then available: application performance. This is due to the radical design of JCOP that has the VM at its core, not as an add-on to a traditional ISO file system card operating system competing with JCOP.
By adding new, and ever faster hardware platforms to the set of supported chips, JCOP consistently grows the headway it has in this crucial customer metric.

	8K EEPROM	16K EEPROM	32K EEPROM	HW DES	Max. RSA key	SHA/MD5	AES	COMP128-1/2	ISO7816 T=0	ISO7816 T=1	ISO14443 T=CL	Mifare support	GlobalPIN	VOP config.	Bio-API
JCOP10		✓	✓	✓	n/a				✓	✓			✓	1	
JCOP20	✓	✓	✓	✓	2048	✓			✓	✓			✓	2	
JCOP30		✓		✓	2048	✓			✓	✓	✓	✓	✓	2	
JCOP21id	✓	✓	✓	✓	2048	✓	✓		✓	✓			✓	3	✓
JCOP21sim	✓	✓	✓	✓	2048	✓	✓	✓	✓				✓	3	
JCOP31bio		✓		✓	2048	✓	✓		✓	✓	✓	✓	✓	3	✓

Functionality
JCOP is available with and without RSA coprocessor support, with and without support for contactless communications, with and without support for GSM protocol features. JCOP also features automatic memory management to run adaptively on chips between 8 and 32k EEPROM.
The public key support includes RSA keys of up to 2048 bit length, the ability to generate all RSA keys on the card for maximum security, as well as the MD5 and SHA1 hashing methods.

Custom Mask Process
IBM has developed a technology process to create transparent blends between any of the JCOP versions and any set of applets into a so-called *Custom Mask*. This way, standard applications of a particular card issuer can be put into the ROM thus reducing the EEPROM requirements significantly. For high-volume rollouts, this can mean substantial savings, as this allows the card issuer to select an 8k EEPROM chip in place of a 32k EEPROM controller that may have had to be used without the JCOP Custom Mask process. Also, the card production time is significantly reduced as it is now no longer necessary to load the standard applets into the EEPROM during card initialization.
This became possible due to the very low-footprint implementation of the JCOP base system, fitting into 32kBytes of ROM in the smallest configuration; consequently, the implementation of a full 2.5G compliant GlobalPlatform JavaCard with SimToolkit functionality, RSA support, and the WIM application all embedded into one mask fits into 64kB of ROM, leaving an additional 32kB of ROM space for card issuer applets on a 32kB EEPROM smart card controller, i.e., overall 64k of applet code and data space.

Cross-industry use
By always integrating the relevant industry standards into the growing list of optional components that can be added to a specific JCOP version, it is possible to serve multiple industries with one technology.
For example, JCOP can be integrated in a version fulfilling the EMV standard when no GSM applet is activated, while being fully ETSI compliant if the SIM Toolkit is activated. Similarly, the JavaCard BioAPI which was primarily intended for use in ID applications, can be made available for financial or telecom use as well to ensure a higher level of assurance of end-user authenticity than is possible via PIN, namely by employing biometric technology.

Quality
IBM JCOP has successfully passed the Visa approval procedures at the highest level 3 (JCOP10, JCOP20, JCOP30). All JavaCard compliance tests are passed by all JCOP versions. All ETSI SimToolkit compliance tests are passed by JCOP21sim. FIPS 140-2 Level 3 certification is to be attained by the end of 2002 (IUT status as of 10/10/02). A Common Criteria evaluation to industry-standard protection profiles is to be completed by the end of 2003.

Low overall card lifetime cost
By the development of technologies for customization of base system and application configuration on low cost chips, JCOP already achieves very low costs of entry to smart card solutions.
In addition to this, the strict adherence to industry standards ensures further cost savings over any proprietary smart card software: All personalization software is standardized, equally standardized card lifecycle management systems can be deployed, thus ensuring a low overall card lifetime cost as compared to any proprietary solution which would require customized personalization and card management software.

IBM's experience
IBM's experience in the implementation of JavaCard/ GlobalPlatform operating systems is extensive: The development started more than 5 years ago, and led to several industry firsts: The first operational OpenPlatform card in use (GSA project, 1998), the first contact-less JavaCard implementation (Cartes, 1999), and the first sub-$3 JCOP version for Visa International (2001), of which more than 20 million copies have been rolled out in 2001 and 2002 alone.

IBM Zurich Research Laboratory
Säumerstr. 4, PO Box
CH-8803 Rüschlikon, Switzerland

Telephone: +41 1 724 8111

The IBM home page can be found on the internet at www.ibm.com and more information on BlueZ Secure Systems and JCOP can be found at www.zurich.ibm.com/Javacard.

IBM is a registered trademark of International Business Machines Corporation.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries. Other company product and service names may be trademarks or service marks of others.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to imply that only IBM's product, program or service may be used. Any functionally equivalent product, program or service may be used instead.

This publication is for general guidance only.

© International Business Machines Corporation 2002.

For further information contact Michael Baentsch, JCOP Product Manager on +41 1 724 8620, or e-mail mib@zurich.ibm.com

JCOP version map

v. 1.1

	Version	8K EEPROM	16K EEPROM	32K EEPROM	HW DES	Max. RSA key length	On-card key-gen.	SHA/MD5	AES	COMP128-1/2	SEED	ISO7816 T=0	ISO7816 T=1	ISO14443 T=CL	Mifare support	Open/Global-Platform version	GlobalPIN	DAP+SD[1]	JavaCard API version	Bio-API	Visa approved	FIPS/140-2 Level 3 [4]	ETSI GSM 03.19 version	ETSI GSM 11.11 version	3GPP TS 03.48 version	3GPP TS 11.14 version	WAP 2.0 WIM
JCOP10	1.0		✓	✓	✓	n/a						✓	✓			2.0.1'[3]	✓		2.1.1		✓						
JCOP20	1.0	✓	✓	✓	✓	1024	✓	✓				✓	✓			2.0.1'[3]	✓		2.1.1		✓						
JCOP30	1.0		✓		✓	1024	✓	✓				✓	✓	✓	✓	2.0.1'[3]	✓		2.1.1		✓						
JCOP10	2.0		✓	✓	✓	n/a						✓	✓			2.0.1'	✓		2.1.1								
JCOP20	2.0	✓	✓	✓	✓	2048	✓	✓				✓	✓			2.0.1'	✓		2.1.1								
JCOP30	2.0		✓		✓	2048	✓	✓				✓	✓	✓	✓	2.0.1'	✓		2.1.1								
JCOP21id	1.0	✓	✓	✓	✓	2048	✓	✓	[2]			✓	✓			2.0.1'	✓	✓	2.1.1	✓		✓					
JCOP21sim	1.0		✓	✓	✓	2048	✓	✓	[2]	✓		✓				2.0.1'	✓	✓	2.1.1				8.3.0	8.0.3	8.8.0	8.5.0	07/01
JCOP31bio	1.0		✓		✓	2048	✓	✓	[2]		[2]	✓	✓	✓	✓	2.0.1'	✓	✓	2.1.1	✓							

For further details on JCOP, please visit http://www.zurich.ibm.com/JavaCard, send an email to javacard@zurich.ibm.com, or contact Michael Baentsch (+41 1 724 8620).

[1] Applet signature and multiple SecurityDomain support as defined in OpenPlatform 2.0.1'
[2] Feature can be activated on demand
[3] according to Visa OpenPlatform Card Implementation Guides; later versions in line with VOP Card Implementation *Requirements*
[4] in evaluation

Inverse JCOP product configuration matrix

v. 1.1

The JCOP configurations on the previous page denote only a subset of the full configuration capabilities of the JCOP family. Use the matrix below to determine which features can be selected for each supported hardware platform to configure a new member of the JCOP family.

Philips Smart Card Controller Reference	DES	Max. RSA key length	On-card key-gen.	SHA/MD5	AES	COMP128-1/2	SEED	ISO7816 T=0	ISO7816 T=1	ISO14443 T=CL	Mifare support	Open/Global-Platform version	GlobalPIN	DAP+SD[1]	JavaCard API version	Bio-API	FIPS/140-2 Level 3[3]	ETSI GSM 03.19	ETSI GSM 11.11	3GPP TS 03.48	3GPP TS 11.14	WAP 2.0 WIM
P8WE6017	✓					✓		✓	✓			2.0.1'	✓	✓	2.1.1							
P8WE6033	✓					✓		✓	✓			2.0.1'	✓	✓	2.1.1							
P8WE5009	✓	2048	✓	✓	✓	✓	[2]	✓	✓			2.0.1'	✓	✓	2.1.1	✓	✓	✓	✓	✓	✓	✓
P8WE5017	✓	2048	✓	✓	✓	✓	[2]	✓	✓			2.0.1'	✓	✓	2.1.1	✓	✓	✓	✓	✓	✓	✓
P8WE5033	✓	2048	✓	✓	✓	✓	[2]	✓	✓			2.0.1'	✓	✓	2.1.1	✓	✓	✓	✓	✓	✓	✓
P8RF5016	✓	2048	✓	✓	✓	✓	✓	✓	✓	✓	✓	2.0.1'	✓	✓	2.1.1	✓	[2]	✓	✓	✓	✓	✓

For further details on JCOP, please visit http://www.zurich.ibm.com/JavaCard, send an email to javacard@zurich.ibm.com, or contact Michael Baentsch (+41 1 724 8620).

[1] Applet signature and multiple SecurityDomain support as defined in OpenPlatform 2.0.1'
[2] Feature can be made available on demand
[3] in evaluation

JCOP Memory Map

v. 1.1

The matrix below shows the amount of memory free for applications either in EEPROM or ROM. The trade-off is flexibility vs. cost: EEPROM code can be loaded post-issuance according to GlobalPlatform; ROM applets must be hard-masked in a standardized procedure. Obviously, ROMed applets do not take up EEPROM space, and thus can save on overall chip cost.

	Chip-ID	EEPROM free	ROM free[1]	Chip ID	EEPROM free	ROM free[1]	Chip ID	EEPROM free	ROM free[1]
JCOP10	P8WE6017	15kB	16kB	P8WE6033	31kB	64kB			
JCOP20	P8WE5009	7kB	24kB	P8WE5017	15kB	24kB	P8WE5033	31kB	56kB
JCOP30	P8RF5016	14kB	24kB						
JCOP21sim	P8WE5017	15kB	-	P8WE5033	31kB	32kB			
JCOP21id	P8WE5009	7kB	20kB	P8WE5017	15kB	20kB	P8WE5033	31kB	52kB
JCOP31bio	P8RF5016	14kB	24kB						

For further details on JCOP, please visit http://www.zurich.ibm.com/JavaCard, send an email to javacard@zurich.ibm.com, or contact Michael Baentsch (+41 1 724 8620).

[1] Each single ROMed applet must not be larger than 16kB.

JCOP10 Technical Brief

Overview: *This document contains a simple overview about the technical capabilities of the first member of the JCOP card OS family, the DES-only version. 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 Guides (http://www.visa.com/nt/suppliers/vendor). All necessary clarifications from ISO7816 and EMV 2000 are also incorporated into the implementation where so required by [1] and [2].

JCOP10v1 is the first member of this family. It conforms to the VOP Card Implementation Guide 2.1.1 Compact from August 2000.

Its successor, JCOP10v2.0 is compliant with the Visa OpenPlatform Card Implementation Requirements Configuration 1, Version 2.0 from February 2002.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default]

ISO7816 T=0 direct convention

ISO7816 T=1 inverse convention

ISO7816 T=0 inverse convention

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

115200 bit/sec

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: 15kB/31kB * (no custom ROM applets).

3.1.2 Transaction buffer

Used to save data written transactional, 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: 651 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communications 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

16kB/64kB * free for applications

* Depending on hardware configurations (see section 5)

4 Supported optional features

Certain features listed in [1] and [2] are not defined to be mandatory. The ones 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 Cryptographic Algorithms

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

Signatures:

ALG_DES_MAC8_NOPAD

ALG_DES_MAC8_ISO9797_M1

ALG_DES_MAC8_ISO9797_M2

RandomData:

ALG_SECURE_RANDOM

KeyTypes:

LENGTH_DES

LENGTH_DES3_2KEY

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

4.2 OpenPlatform

4.2.1 Global PIN

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

5 Supported Hardware

The identical JCOP mask can run on these platforms and automatically make use of their resources, i.e., create a bigger persistent Java heap, and make more ROM memory available for ROMed applets*.

5.1 Philips P8WE6017

48 kB ROM

16 kB EEPROM

1300 Bytes RAM

Triple-DES coprocessor

5.2 Philips P8WE6033

96 kB ROM

32 kB EEPROM

1300 Bytes RAM

Triple-DES coprocessor

* Custom applets may be submitted to Philips for inclusion into a ROM mask. The maximum package size is 16kB.

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 3.71 MHz. The applets are the Visa approved versions after having been initialized and populated with keys as required for Visa VTF 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 *	162	15.1
INIT UPDATE CardManager	538	50.0
EXTERNAL AUTH CardManager	327	30.4
Install VisaCash	8025	746.3
SELECT VisaCash*	162	15.1
Initialize LOAD for VisaCash	833	77.5
Perform LOAD for VisaCash	1587	147.6
Initialize PURCHASE for VisaCash	222	20.6
Perform PURCHASE for VisaCash	1629	151.5
ReadBalance from VisaCash	96	8.9
SELECT VSDC*	582	54.1
GenerateAC (ARQC) from VSDC	2459	228.7

*. Not first SELECT to eliminate potential applet setup effects

A Revision History

1.0 Initial document

1.1 Correction of typos

1.2 AIDs of ROMed applets added

1.3 References to external specifications added

1.4 Incorrect reference to PseudoRandom constant deleted (5.1.2)

1.5 Document reformatted

1.6 Timing explanations added (6)

2.0 Specification update to cover JCOP10v2.0 (1); correction of RAM heap size (3.2.1)

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/

JCOP20 Technical Brief

Overview: *This document contains a simple overview about the technical capabilities of the first PK 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 Guides (http://www.visa.com/nt/suppliers/vendor). All necessary clarifications from ISO7816 and EMV 2000 are also incorporated into the implementation where so required by [1] and [2].

JCOP20 is the public-key member of this family. It conforms to the VOP Card Implementation Guide 2.1.2 Compact from August 2000. It adds public key (RSA) capabilities to the ones listed in JCOP10.

Its successor, JCOP20v2.0 is compliant with the Visa OpenPlatform Card Implementation Requirements Configuration 2 with PK, Version 2.0 from February 2002.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default]

ISO7816 T=0 direct convention

ISO7816 T=1 inverse convention

ISO7816 T=0 inverse convention

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

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: 7kB/15kB/31kB* (without custom ROM applets).

3.1.2 Transaction buffer

Used to save data written transactional, 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: 615 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communications 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

24kB/56kB* free for applications

* Depending on hardware configurations (see section 5)

4 Supported optional features

Certain features listed in [1] and [2] are not defined to be mandatory. The ones 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 Cryptographic Algorithms

JCOP20 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* (v2 only)

LENGTH_RSA_1024*

LENGTH_RSA_768*

LENGTH_RSA_512*

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

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.

* 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.

5 Supported Hardware

The identical JCOP mask can run on these platforms and automatically make use of their resources, i.e., create a bigger persistent Java heap, and make more ROM memory available for ROMed applets*.

5.1 Philips P8WE5017

64 kB ROM

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.2 Philips P8WE5033

96 kB ROM

32 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.3 Philips P8WE5009

64 kB ROM

8 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.

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 3.71 MHz. The applets are the Visa approved versions after having been initialized and populated with keys as required for Visa VTF 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 *	145	13.5
INIT UPDATE CardManager	469	45.0
EXTERNAL AUTH CardManager	302	28.1
Install VisaCash	4928	458.3
SELECT VisaCash*	154	14.3
Initialize LOAD for VisaCash	769	71.5
Perform LOAD for VisaCash	1453	135.1
Initialize PURCHASE for VisaCash	207	19.3
Perform PURCHASE for VisaCash	1479	137.6
ReadBalance from VisaCash	91	8.5
SELECT VSDC*	549	51.1
GenerateAC from VSDC	2282	212.3

The PK operations are largely dominated by the hardware speed of the FameX. 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
Generate 1024bit CRT key	~ 5900
1024 bit CRT public key operation (F4)	50
1024 bit CRT private key operation	250

* Not first SELECT to eliminate potential applet setup effects

A Revision History

0.9 Preliminary version (prior to Visa approval)

0.91 AIDs or ROMed applets added

0.92 References to external specifications added

0.93 Incorrect reference to PseudoRandom constant deleted (4.1.2)

1.0 Document reformatted

1.1 Timing explanations added (6); correction of size information (3.3, 5.1)

1.2 Information on supported RSA key lengths added (4.1.2); Information an additional platform support added (3.1, 5.3)

2.0 Specification update to cover JCOP20v2 (1, 4.1.2); correction of RAM heap size (3.2.1); added previously omitted KeyPair constant (4.1.2)

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/

JCOP30 Technical Brief

***Overview**: This document contains a simple overview about the technical capabilities of the first contact/contactless (dual-interface) 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 Guides (http://www.visa.com/nt/suppliers/vendor). All necessary clarifications from ISO7816 and EMV 2000 are also incorporated into the implementation where so required by [1] and [2].

JCOP30 is the first dual-interface member of this family. It conforms to the VOP Card Implementation Guide 2.1.3 Compact from August 2000.

Its successor, JCOP30v2.0 is compliant with the Visa OpenPlatform Card Implementation Requirements Configuration 2 with PK, Version 2.0 from February 2002.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default]

ISO7816 T=0 direct convention

ISO7816 T=1 inverse convention

ISO7816 T=0 inverse convention

ISO14443A T=CL

2.2 Supported speeds

2.2.1 Contact protocols:

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

2.2.2 Contactless protocol:

106000 bit/sec

212000 bit/sec

424000 bit/sec

848000 bit/sec*

* not approved for use in production cards

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: 14kB (without custom ROM applets). Only for JCOP30v1: 13kB

3.1.2 Transaction buffer

Used to save data written transactionally, 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: 523 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communications 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

24kB free for applications*

* Custom applets may be submitted to Philips for inclusion into a ROM mask (see section 5)

4 Supported optional features

Certain features listed in [1] and [2] are not defined to be mandatory. The ones 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 Cryptographic Algorithms

JCOP30 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* (JCOP30v2 only)

LENGTH_RSA_1024

LENGTH_RSA_768

LENGTH_RSA_512

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

TYPE_RSA_CRT_PRIVATE

KeyPair:

ALG_RSA_CRT

4.1.3 APDU class

The method APDU.getProtocol() returns according to [1] the currently activated communications protocol. In compliance with [1], JCOP30 returns APDU.PROTOCOL_T0 (0) if T=0 is running, and APDU.PROTOCOL_T1 (1) if T=1 is running. JCOP30 returns none of these constants if T=CL is running. Hence, using a query of the form

```
if ((APDU.getProtocol() != APDU.PROTOCOL_T0) && (APDU.getProtocol() != APDU.PROTOCOL_T1))
```

can be used to cease computation in an applet that does not wish to execute if run over the contactless interface.

4.2 OpenPlatform

4.2.1 Global PIN

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

* 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.

5 Supported Hardware

The supported configuration includes a 1kB Mifare Emulation mode (“Mifare Standard”) ensuring interoperability of JCOP30 in existing Mifare infrastructures. The usual JCOP ROM applet integration faciliti es can be used to create custom masks*.

5.1 Philips P8RF5016

64 kB ROM

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

Mifare Standard Emulation

* 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 JCOP30v2.

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 3.71 MHz. The applets are the Visa approved versions after having been initialized and populated with keys as required for Visa VTF 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 *	82	7.6
INIT UPDATE CardManager	276	25.7
EXTERNAL AUTH CardManager	191	17.8
Install VisaCash	4198	390.4
SELECT VisaCash*	88	8.2
Initialize LOAD for VisaCash	416	38.7
Perform LOAD for VisaCash	895	83.2
Initialize DEBIT for VisaCash	116	10.8
Perform DEBIT for VisaCash	894	83.1
ReadBalance from VisaCash	52	4.8
SELECT VSDC*	182	16.9
GenerateAC from VSDC	1609	149.6

The PK operations are largely dominated by the hardware speed of the FameX. 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
Generate 1024bit CRT key	~ 5000
1024 bit CRT public key operation (F4)	32
1024 bit CRT private key operation	220

* Not first SELECT to eliminate potential applet setup effects

A Revision History

0.9 Preliminary version (prior to Visa approval)

0.91 AIDs or ROMed applets added

0.92 References to external specifications added

0.93 Incorrect reference to PseudoRandom constant deleted (4.1.2)

1.0 Document reformatted

1.1 Timing explanations added (6); ROM masking sentence added (5)

1.2 Information on supported RSA key lengths added (4.1.2)

1.3 Information on free ROM updated (3.3) and Information on APDU class and usage of T=CL added (4.1.3)

2.0 Specification update to cover JCOP30v2 (1, 4.1.2, 3.1.1); correction of RAM heap size (3.2.1, 1); caveat on T=CL speed added (2.2.2); added previously omitted KeyPair constant (4.1.2)

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/

JCOP21sim Technical Brief

Overview: *This document contains a simple overview about the technical capabilities of the first GSM SIM 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 Guides (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].

JCOP21sim is the first GSM SIM member of this family. It adds GSM SIM and WAP WIM capabilities according to [3-9] to the ones listed in JCOP20.

JCOP21sim can also be configured to be fully compliant to EMV for use as a banking card if the GSM/WAP functionality is not activated.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Javabased trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=0 direct convention [default]

ISO7816 T=0 inverse convention

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

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* (without custom ROM applets).

3.1.2 Transaction buffer

Used to save data written transactional (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 communications 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

0/32kB* free for applications

* Depending on hardware configurations (see section 5)

4 Supported optional features

Certain features listed in the standards implemented are not defined to be mandatory. The ones 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

JCOP21sim 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*

LENGTH_RSA_1024

LENGTH_RSA_768

LENGTH_RSA_512

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

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

JCOP21sim allows the installation of multiple Security Domains as well as Mandated DAP verification.

4.3 GSM

JCOP21sim fully implements the GSM SIM functionality as defined in [3], [4] and [6]. Furthermore, it provides the SIM API for Java Card [5], which allows the installation of SIM Toolkit applications written in Java. Also, the security mechanisms defined in [7] are fully implemented allowing over-the-air loading of Java applications as well as remote file-system management.

By default JCOP21sim supports the COMP128 (version 1 and 2) cryptographic algorithm for GSM authentication. Other algorithms can be integrated on demand.

* 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.

4.4 WAP

The JCOP21sim system provides a built-in WAP 2.0 WIM application [8], which can be optionally installed when the SIM functionality is activated. This allows configuring JCOP21sim as SIM-only or as SWIM card. The WIM application supports application-level signatures as well as WTLS security based on 1024-bit RSA operations and the WTLS SHA-1 pseudo-random function as defined in [9].

5 Supported Hardware

The identical JCOP mask can run on these platforms and automatically make use of their resources, i.e., create a bigger persistent Java heap, and make more ROM memory available for ROMed applets*.

5.1 Philips P8WE5017

64 kB ROM

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.2 Philips P8WE5033

96 kB ROM

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.

A Revision History

1.0 Initial document

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] GSM 11.11: Specification of the Subscriber Identity Module 8.0.3

[4] 3GPP TS 11.14: Specification of the SIM Application Toolkit 8.5.0

[5] 3GPP TS 03.19: Subscriber Identity Module API for JavaCard 8.3.0

[6] GSM 03.40: Technical Realization of the Short Message Service (SMS) 7.4.0

[7] 3GPP TS 03.48: Security Mechanisms for SIM Application Toolkit 8.8.0

[8] WAP-260-WIM-20010712-a: Wireless Application Protocol, Wireless Identity Module

[9] WAP-261-WTLS-20010406-a: Wireless Application Protocol, Wireless Transport Layer Security

JCOP31bio Technical Brief

***Overview**: This document contains a simple overview about the technical capabilities of the first biometry-enabled contact/contactless (dual-interface) 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 so required by [1] and [2].

JCOP31bio is the first biometry-enabled dual-interface member of this family. It conforms to the VOP Card Implementation Requirements Configuration 3 with PK, Version 2.0 from February 2002.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default]

ISO7816 T=0 direct convention

ISO7816 T=1 inverse convention

ISO7816 T=0 inverse convention

ISO14443A T=CL

2.2 Supported speeds

2.2.1 Contact protocols:

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

2.2.2 Contactless protocol:

106 kbit/sec [default]

212 kbit/sec

424 kbit/sec

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: 13 kB (without custom ROM applets and with one biometric algorithm in EEPROM).

3.1.2 Transaction buffer

Used to save data written transactionally, 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: 523 bytes

3.2.2 APDU buffer

Used to hold incoming and outgoing communications 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

24kB free for applications*

* Custom applets may be submitted to Philips for inclusion into a ROM mask (see section 5)

4 Supported optional features

Certain features listed in [1] and [2] are not defined to be mandatory. The ones 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 Cryptographic Algorithms

JCOP31bio 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*

LENGTH_RSA_1024

LENGTH_RSA_768

LENGTH_RSA_512

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

TYPE_RSA_CRT_PRIVATE

KeyPair:

ALG_RSA_CRT

4.1.3 APDU class

The method APDU.getProtocol() returns according to [1] the currently activated communications protocol. In compliance with [1], JCOP31bio returns APDU.PROTOCOL_T0 (0) if T=0 is running, and APDU.PROTOCOL_T1 (1) if T=1 is running. JCOP31bio returns none of these constants if T=CL is running. Hence, using a query of the form

```
if ((APDU.getProtocol() != APDU.PROTOCOL_T0) && (APDU.getProtocol() != APDU.PROTOCOL_T1))
```

can be used to cease computation in an applet that does not wish to execute if run over the contactless interface.

4.2 OpenPlatform

4.2.1 Global PIN

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

4.2.2 Multiple SecurityDomains / DAP

JCOP31bio allows the installation of multiple Security Domains as well as Mandated DAP verification.

* 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.

4.3 Biometry

4.3.1 Scope

JCOP31bio implements the full biometry API as defined in [3].

4.3.2 Supported constants

Different on-card biometric match algorithms can be made available for JCOP31bio. The first one supported is a fingerprint matching algorithm. The supported API constant in the API thus is

org.javacardforum.javacard.biometry.BioBuilder

FINGERPRINT

5 Supported Hardware

The supported configuration includes a 1kB Mifare Emulation mode (“Mifare Standard”) ensuring interoperability of JCOP31bio in existing Mifare infrastructures. The usual JCOP ROM applet integration facilities can be used to create custom masks*.

5.1 Philips P8RF5016

64 kB ROM

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

Mifare Standard Emulation

* Custom applets may be submitted to Philips for inclusion into a ROM mask. The maximum package size is 16kB.

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]	79	7.3
INIT UPDATE CardManager [1]	293	27.2
EXTERNAL AUTH CardManager [1]	187	17.4
Install VisaCash [2]	3868	359.7
SELECT VisaCash [1]	88	8.8
Initialize LOAD for VisaCash [1]	438	40.7
Perform LOAD for VisaCash [1]	896	83.3
Initialize DEBIT for VisaCash [1]	122	11.3
Perform DEBIT for VisaCash [1]	984	91.5
ReadBalance from VisaCash [1]	58	5.4
SELECT VSDC [1]	102	9.5
GenerateAC from VSDC [1]	1436	133.5

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. Also note that two figures are given: The first for typical contact-less performance (slower, requiring less power consumption during contactless operation), one for typical contact-only operation (more power, hence faster).

Operation	Msec
1024 bit CRT public key operation (F4)	24 / 18
1024 bit CRT private key operation	309 / 163
Generate 1024bit CRT key	~7900 / ~4200
Generate 2048bit CRT key	~83000 / ~47000

[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

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] JavaCard Forum: Biometry API specification: http://www.javacardforum.org/Documents/Biometry/biometry.html

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.

JCOP, BlueZ and all BlueZ-based trademarks and logos are trademarks or registered trademarks of International Business Machines Corp. in the US and other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the US and other countries.

2 Communications

2.1 Supported protocols

ISO7816 T=1 direct convention [default]

ISO7816 T=0 direct convention

ISO7816 T=1 inverse convention

ISO7816 T=0 inverse convention

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

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 beenimplemented 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*

LENGTH_RSA_1024

LENGTH_RSA_768

LENGTH_RSA_512

TYPE_DES_TRANSIENT_RESET

TYPE_DES_TRANSIENT_DESELECT

TYPE_DES

TYPE_RSA_PUBLIC

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 FIPS 140-2 (rf. http://csrc.ncsl.nist.gov/cryptval). The complete system with a PKCS#15 applet is scheduled to receive FIPS140-2 level 3 certification by April 2003.

* 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.

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

64 kB ROM

8 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.2 Philips P8WE5017

64 kB ROM

16 kB EEPROM

2300 Bytes RAM

Triple-DES coprocessor

FameX RSA coprocessor

5.3 Philips P8WE5033

96 kB ROM

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.

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

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.