SENTINELS

A Comparison of Java Cards: State-of-Affairs 2006*

Wojciech Mostowski[1] Jing Pan[2] Srikanth Akkiraju[3]

Erik de Vink[2] Erik Poll[1] Jerry den Hartog[3]

Abstract

This document presents the results of a comparative study of some popular Java Cards on the market. Eight different cards from four manufacturers have been considered. The analysis has been done at two levels – (i) a documentation-based comparison, also taking other publicly available resources into account, (ii) an actual hands-on testing with software developed specifically for this purpose by the PinpasJC research team. The investigations focus on basic functionality, secure channels, the transaction mechanism, support symmetric and asymmetric cryptography, Global Platform and Open Platform compliance, and garbage and memory management.

1 Introduction

Java Card plays an increasingly predominant role in smart card projects, e.g. for identity cards and travel documents. Many vendors respond to this market expansion with dedicated products. However, by design, these products are not exactly equivalent. On top of the traditional dissimilarities such as component size, many behavioural differences can be detected both at the functional and performance level. As such, this can have impact on the portability of a solution and undermine the advantage of using Java Card.

This document presents a comparative analysis of eight commercial Java Cards available to us to date (Autumn/Winter 2006), namely `C_211A`, `C_211B`, `B_211`, `B_22`, `B_221`, `A_211`, `A_221`(two slightly different instances differing in the communication speed), and `D_22`. The evaluation took place with respect to the following criteria:

*This work is supported by the research program Sentinels (`http://www.sentinels.nl`). Sentinels is financed by the Technology Foundation STW, the Netherlands Organisation for Scientific Research NWO, and the Dutch Ministry of Economic Affairs.

[1]Computing Science Department, Radboud University, Nijmegen, The Netherlands

[2]Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands

[3]Department of Electrical Engineering, Mathematics and Computer Science, Twente University, Enschede, The Netherlands

Compliance to standards: Compatibility to Java Card and Global Platform

Implemented features: Communication interfaces, APDU protocols, memory management, atomicity, RMI support and on-card byte-code verification availability

Performance: Execution time of cryptographic algorithms

Limits: Transaction commit capacity and APDU buffer capacity

The investigations discussed below, give an indication of the present state of affairs regarding Java Card. For example, the choice of cards under consideration has been limited by their availability. It is also noted that some results reported in this document have been based on publicly available information only. Additionally, when it came to actual card testing, two major bottlenecks were hindering our progress for some time. First of all, getting hold of a type of Java Card in small quantities is non-trivial. Secondly, some cards are personalized with proprietary authentication keys (or, more precisely, keys that are derived following proprietary schemes). Finding out this information took a considerable amount of time (see also Section 3.1).

The rest of this document is organized as follows. Section 2 discusses the card features stated in the cards' documentations [16, 15, 2, 1, 6], regarding the four evaluation criteria above. Section 3 presents several tests (and their results) we performed with the cards to explore various features: basic card features, secure channel functionality, transaction mechanism, cryptography support and speed, RMI, garbage collection, etc. For each of the tests the methodology is briefly described. Finally, Section 4 concludes the report.

2 Card Features Based on Documentation

The Java Cards that have been considered in the research reported here are the following:

- From Manufacturer C the `C_211A` and `C_211B` cards. These cards are the only cards that we were able to buy directly from the manufacturer in small quantities. They are also the only cards that have full technical documentation that is publicly available regarding the particular Global Platform implementation. On the other hand Manufacturer C continuously sells cards known to have bugs, see comments in Section 3.8.

- From Manufacturer B the following cards: an older `B_211`, `B_22`, and `B_221`. The `B_221` card is the most advanced (supporting both Java Card 2.2.1, Global Platform 2.1.1 and contactless interface) card from Manufacturer B currently available.

- From Manufacturer A the `A_211` card and `A_221` card. For the `A_221` card we have two different instances available, the main difference being the communication speed of the contact interface. Whenever any substantial difference has been noticed between the two `A_221` cards we noted them down. Also, `A_221` is the only other card in our test set that supports the latest Java Card and Global Platform technologies.

Java Card	Company	JC API	Open/Global Platform
C_211A	Manufacturer C	2.1.1	2.0.1
C_211B	Manufacturer C	2.1.1	2.0.1
B_211	Manufacturer B	2.1.1	2.0.1
B_22	Manufacturer B	2.2	2.0.1
B_221	Manufacturer B	2.2.1	2.1.1
A_211	Manufacturer A	2.1.1	2.0.1
A_221	Manufacturer A	2.2.1	2.1.1
D_22	Manufacturer D	2.2	2.1.1

Table 1: Compliance to software standards

Java Card	EEPROM(KB)	RAM(Bytes)	ROM(KB)
C_211A	32 (30)	4096	96
C_211B	64 (—)	—	—
B_211	32 (29)	—	—
B_22	64	—	—
B_221	16/32/64	—	—
A_211	32 (30)	2300	96
A_221	72 (70)	4608	160
D_22	64	—	—

Table 2: Memory characteristics

- From Manufacturer D the D_22 card.

2.1 Compliance to Standards

Table 1 compares the cards under consideration from the point of view of specific versions of the Java Card API and Open/Global Platform standard that they support.

Table 2 provides the hardware features of the cards. The values between parentheses are the amounts of free memory available for applications once the system is loaded. The documentation from the vendor of B_211 card does not show any information about the capacity of RAM or ROM. The total amount of ROM size of the A_221 card is not presented in any documentations either,[1] though the vendor of this card's microcontroller does say that the card has 160KB of ROM.

2.2 Implemented Features

Table 3 compares the cards under consideration with respect to the data transport layer, whereas Table 4 provides an overview of the availability of additional features such as

[1]There is actually no formal documentation available for this card as other cards from Manufacturer A.

Java Card	APDU Protocols	Communication Interface
C_211A	T=0	Contact
C_211B	T=0, T=1	Contact
B_211	T=0, T=1	Contact
B_22	T=0, T=1	Contact
B_221	T=0, T=1, T=CL	Contact, Contactless
A_211	T=0, T=1	Contact
A_221	T=0, T=1, T=CL	USB 2.0 (Low Speed) Contact, Contactless
D_22	T=0, T=1, T=CL	Contact, Contactless, USB

Table 3: Communication features

Java Card	Garbage Collection	RMI Supported	On-card Byte-code Verification	Logical Channel
C_211A	—	No	Yes	No
C_211B	—	No	Yes	No
B_211	Yes	No	—	No
B_22	Yes	Yes	—	Yes
B_221	Yes	Yes	—	Yes
A_211	Full	No	—	No
A_221	Full	Yes	—	Yes
D_22	—	Yes	—	—

Table 4: Java Card and Open Platform features

garbage collection, RMI, on-card byte-code verification and logical channels. Availability of many of these cannot be decided given the information in the vendors' documentations. For example, Manufacturer B claims that their cards support run-time garbage collection. However, it remains implicit whether they concern full or partial garbage collection. Both cards from Manufacturer A supposedly provide full garbage collection (see Section 3.11 on garbage collection). Note that logical channel functionality is added only to Global Platform specification 2.1.1 as an optional feature.

2.3 Performance

As far as we are aware, there is no publication available that actually considers a performance comparison for Java Cards. An exception is [4] which, at present, treats rather outdated cards. Manufacturer A has certain documents for each of their cards (excluding A_221), where a list of performance figures can be found, though qe are not in a position to confirm these figures. Recently, a project to measure smart card performance has been initiated in France,[2] but at the moment the project is in its very early stage.

[2] http://cedric.cnam.fr/mesure/

Java Card	Transaction Buffer Size (bytes)	APDU Buffer Size (bytes)
C_211A	—	255
C_211B	—	—
B_211	—	—
B_22	—	—
B_221	—	—
A_211	512	261
A_221	—	—
D_22	—	—

Table 5: Buffer capacity limits

2.4 Limits

The limits we consider are the size of the APDU buffer and the transaction commit buffer. The APDU buffer is used to hold incoming and outgoing communication data. The transaction buffer is used to save data involved in transactions, viz. all persistent byte and short stores, as well as persistent parameters to `Util.arrayCopy`. Only a few vendors have the buffer size figures in their documentations. However, some of these figures can be retrieved directly from the card through the Java Card API, see Section 3. The figures obtained from the documentation (only a few) are listed in Table 5.

Some of the limits are not documented at all (we mention the maximum number of Java Card objects managed, maximum size of applets or load files, maximum number of load files or applets that can be installed on a card, maximum number of secure channel keys in the Issuer Security Domain).

The data we have presented so far are based solely on available documentation (if any). In the next section we describe a number of tests we performed to verify the documented data and to obtain details not included in the documentation.

3 Card Testing

For the purpose of testing a number of applications and applets have been written. We have tested some basic features (Section 3.2), Global Platform functionality, in particular secure channels (Section 3.3), the transaction mechanism (Section 3.4). Furthermore, we have established the range of cryptographic support on the cards. We also performed speed and compatibility tests (Section 3.5) and put an effort to test the basic RMI functionality of RMI enabled cards (Section 3.7), to test the GP API support (Section 3.8) and garbage collection (Section 3.11).

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.