

Computing with Java Cards^{TM*}

Serge CHAUMETTE

Pascal GRANGE

Damien SAUVERON^{†‡}

Pierre VIGNERAS

{serge.chaumette,pascal.grange,damien.sauveron,pierre.vigner}@labri.fr

LaBRI, Laboratoire Bordelais de Recherche en Informatique

UMR 5800 – Université Bordeaux 1

351 cours de la Libération, 33405 Talence CEDEX, FRANCE.

ABSTRACT

More and more pieces of hardware are getting connected to the Internet every day. Technologies such as Bluetooth or Wi-Fi make this process even faster. All these equipments make sense provided they can communicate with each other. Among the communication paradigms that seem suited for such an environment are mobile codes or mobile agents and/or remote procedure calls. These imply executing a code that is either coming from somewhere over the network, or that is local but managed remotely like it is the case for the grid, for instance. Security is then one of the main challenges that has to be dealt with. The aim of this paper is to present a Java Card based platform that we are in the process of setting up to experiment this challenge.

KEYWORDS: *Smart card, Java Card, distributed computing, security, grid.*

1. INTRODUCTION

One of the reasons for setting up a grid [1, 2] or for connecting computing resources together is to allow people or companies to use computing units provided by third parties. The problem is that the

*Java and all Java-based marks are trademarks or registered trademarks of Sun microsystems, Inc. in the United States and other countries. The authors are independent of Sun microsystems, Inc.

[†]LaBRI and ITSEF center of SERMA Technologies.

[‡]This work is partly achieved in the framework of a doctoral grant from the french ministry of research and SERMA Technologies.

user has to trust the owner of the computer where his code will be executed. Even though some level of software security can be provided, nothing can prevent the user from malicious operations on its program which could result from a physical access to the computation unit: confidential data can be eavesdropped, calculation can be disturbed and results can be tampered with.

We believe that the use of smart cards [3] can make it possible to cope with these problems. The physical protection that they offer ensures that it will be infeasible, in a reasonable amount of time, to understand what is stored and what internally occurs. This is not the case with a traditional processor.

In this paper we present a project that we are setting up, the aim of which is to implement a cluster of Java Cards and a programming framework on top of it. We claim that this will enable to define a general approach to provide secure computing on third party hardware.

2. STATE OF THE ART

It is now acknowledged that the ever increasing computing power of smart cards should allow to achieve some effective calculation [4] even though it is clear that high performance cannot be expected.

Frameworks such as JiniCard [6], Jason [5] or OrbCard [7] have been developed to communicate in transparent and secure ways with the services offered by smart cards. Although these do not directly consider distributed computing with smart cards as their target paradigm, it is still possible to use them to achieve this goal. Of course they miss features that

are useful in the context of distributed application using smart card, for instance, asynchronous method invocation [9, 10].

More formal approaches to protect mobile codes executed on untrusted runtime environments have been developed. For instance, there is an original solution based on an extension of function hiding using error correcting codes [8]. These solutions, although supported by strong foundations, seem to be difficult to use in practice due to the assumptions they make on the applications.

3. OUR PROJECT

Based on the multi-applicative feature of the Java Card technology [11, 12] and on our experience in both this technology and the technologies of distributed computing, we consider possible to set up a hardware platform, *i.e.* a cluster of smart cards, and to provide a framework for developing and managing applications on this cluster.

We understand a *software framework* as the APIs for the developer and the tools for the end-user or the administrator. This framework will be based on some pre-existing system such as RMI [13], JavaParty [14], JiniCard [6], Jason [5], OrbCard [7] or Mandala [15, 16]. Mandala is a general framework that we have developed for distributed computing. It provides an RMI-like abstraction. Mandala offers features that we believe are useful in the context of this work such as the *active container* [17, 18] concept it is based on, or the *asynchronism* it provides for remote method invocation.

The hardware platform is presented in figure 1. The smart cards all together make up some sort of grid or more precisely of cluster. They are powered by the smart card readers themselves. These USB readers are chained together and connected to a certain number of hosts which will be used to manage them.

As our framework is Java-based, we will use the OpenCard Framework [19] and a bridge from OCF to PC/SC, the Personal Computer/Smart Card standard [20], or JPC/SC [21], a JNI-wrapper for PC/SC, to control the readers and the deployment of the applications (*cf.* figure 2). PC/SC is a standard which provides a high level API to communicate with smart

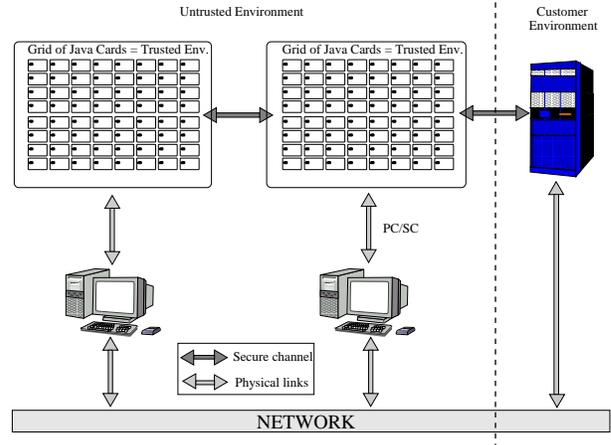


Figure 1: A solution for computing with Java Cards

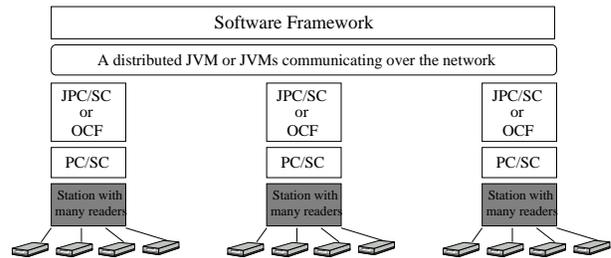


Figure 2: The software framework

card readers. Each PC/SC compatible reader comes with a pluggable driver used to communicate with the PC/SC middleware (*cf.* figure 3). Using only PC/SC compatible readers makes it possible to use readers of different suppliers without deep knowledge of the underlying specific protocols.

There is also the recent OpenCT [22] project allowing the support of the readers under Linux but it is still in version alpha and it is not a standard.

4. MAIN CHALLENGES

We have identified the following challenges to cope with in order to achieve secure computing on our cluster of Java Cards:

- The memory size of Java Cards. To handle this constraint we could for instance cipher and store the intermediate results in a standard unsecure memory that offers a large capacity and a

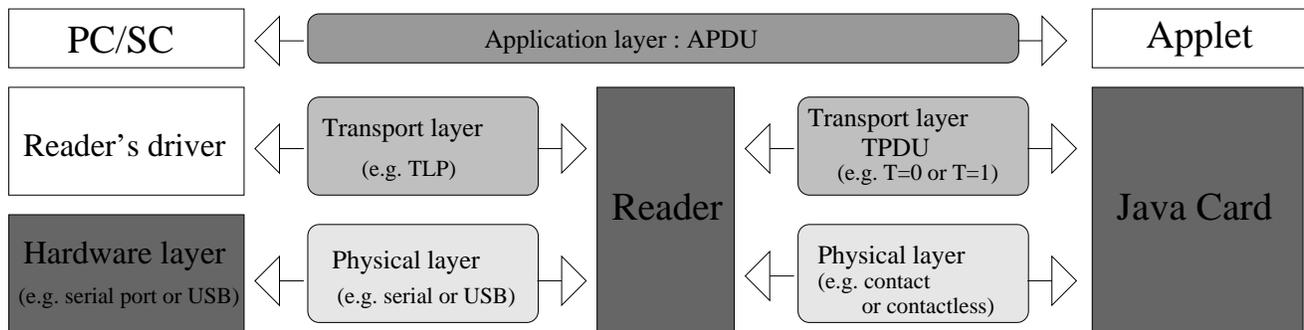


Figure 3: The PC/SC solution

fast access. We could also use a virtual memory distributed over a number of dedicated smart cards, what would make it possible to keep the data secret, but we still would have to cipher the data to be transferred through the insecure channels between the cards.

- Insecure channels. The communications will be ciphered between the clients and the grid of smart cards, and within the grid itself between the smart cards.
- Execution model. To be efficient, our framework will propose asynchronism and an interactive mode of operation allowing the smart cards to behave as a server or a client.
- Heterogeneity and deployment of applications. To be usable our framework will be transparent, to manage the heterogeneity of the hardware (smart cards, readers and hosts), and to provide fault-tolerance mechanisms. It will also be necessary to manage the deployment, the termination and the mapping of the codes over the cards.

5. HARDWARE CONSIDERATIONS

The OS of the platform will be Linux because the middleware PC/SC layer under Windows (for example Windows 98) supports only one reader at the same time. By now the Linux PC/SC middleware layer only supports for the moment 16 readers but the adaptation for more readers does not seem to be difficult.

To we wish to use readers which sources are free. As far as we know there are no USB readers with an open source driver for windows and those we have identified for Linux are not sold any longer.

Furthermore, we would like to use USB CCID [23] readers. The aim of this standard is to allow various CCID readers from different manufacturers to be supported by the same driver. But, at the present time, there is not any free driver for these readers under Linux.

6. PERSPECTIVES

By coping with these challenges we will gain experience on secure computing over a grid of Java Cards. We believe that we will then be able to apply this experience to any piece of equipment that can be connected to a network and that provides Java Card-like hardware level security.

7. REFERENCES

- [1] Fran BERMAN, Anthony J.G. HEY and Geoffrey FOX.
Grid Computing: Making The Global Infrastructure a Reality.
John Wiley & Sons, 2003 - ISBN 0-470-85319-0
- [2] Ian FOSTER and Carl KESSELMAN.
The Grid: Blueprint for a New Computing Infrastructure.
Morgan Kaufmann, 1999 - ISBN 1-558-60475-8
- [3] Wolfgang RANKL and Wolfgang EFFING.
Smart Card Handbook 2nd edition.
John Wiley & Sons, 2000 - ISBN 0-471-98875-8

- [4] Christoph SIEGELIN, Laurent CASTILLO and Ulrich FINGER.
Smart cards: distributed computing with \$5 Devices.
Parallel Processing Letters, Vol.11, N°1 (2001) 57-64.
- [5] Richard BRINKMAN and Jaap-Henk HOEPMAN.
Secure Method Invocation in JASON.
- [6] Roger KEHR, Michael ROHS and Harald VOGT.
Issues in Smarcard Middleware.
- [7] Alvin T.S. CHAN, Florine TSE, Jiannong CAO and Hong VA LEONG.
Enabling Distributed Corba Access to Smart Card Applications.
IEEE Internet Computing, pp.27-36, May/June 2002.
- [8] Sergio LOUREIRO and Refik MOLVA.
Mobile Code Protection with Smarcards.
- [9] Didier DONSEZ, Sebastien JEAN, Sylvain LECOMTE and Olivier THOMAS
(A)synchronous Use of Smart Cards Services Using SOAP and JMS
- [10] Sebastien JEAN, Didier DONSEZ and Sylvain LECOMTE.
Smart Card Integration in Distributed Information Systems: the Interactive Execution Model.
- [11] Sun microsystems.
Java Card™ 2.2 Specifications.
<http://java.sun.com/products/javacard/>
- [12] Zhiqun CHEN.
Java Card™ Technology for Smart Cards.
Addison-Wesley - ISBN 0-201-70329-7.
- [13] William GROSSO.
Java RMI.
O'Reilly & Associates, Inc., 2002 - ISBN 1-56592-452-5
- [14] Michael PHILIPPSEN and Matthias ZENGER.
JavaParty: Transparent remote objects in Java. Concurrency: Practice and Experience, 9(11):1225-1242, November 1997.
- [15] Pierre VIGNERAS and Pascal GRANGE
Mandala.
<http://mandala.sourceforge.net/>
- [16] Serge CHAUMETTE and Pierre VIGNERAS
A framework for seamlessly making object oriented applications distributed.
Parallel Computing 2003.
Dresden, Germany, September 2-5, 2003.
- [17] Pierre VIGNERAS.
JACOb: a software framework to support the development of e-services, and its comparison to Enterprise JavaBeans.
International Workshop on Performance-Oriented Application Development for Distributed Architectures (PADDA) 2001.
- [18] Serge CHAUMETTE and Pierre VIGNERAS.
Active containers: an alternative approach to mobile agents systems.
Second International Symposium on Object Oriented Parallel Environments, ISCOPE 98.
Santa Fe, NM, USA. Poster
- [19] OpenCard Framework.
<http://www.opencard.org/>
- [20] PC/SC Specifications.
<http://www.pcscworkgroup.com/>
- [21] The JPC/SC specifications and driver.
<http://www.linuxnet.com/middleware/>
- [22] OpenCT.
<http://www.opensc.org/cvs/openct/>
- [23] Chip/Smart Card Interface Devices (CCID).
<http://www.usb.org/developers/>