Smart Card Solution: Highly secured Java Card Technology

Surender Reddy Adavalli
Department of Computer Science, University of Auckland
sada027@ec.auckland.ac.nz

Abstract

Smart card is a tiny personal computer without a screen and keyboard. Smart card is tamper-resistant hardware with high system level security to secure internal procedures on the chip. Along with this it also needs secured operating system and application software to make illegal access to the data impossible. Java card technology satisfies all the requirements for development of smart card solutions. In this paper, we will concentrate on application software requirements for smart card, security issues and how Sun Java Card becomes the ultimate smart card solution.

1. Introduction

Smart card also called as a microprocessor card has found its place in enormous number of applications like electronic banking, access control, mobile phones (SIM) etc., with its increasing role in real time applications and hence demands reliable and highly secured application development software. Smart card provides different levels of security ranging from simple building access control to complex multi-bank transactions. In early day of smart card evolution, smart card applications were developed depending on the chip specifications using proprietary software and then chip manufacturers used to embed the application into the chip. This application cannot be modified once it is embedded into the chip as it is burnt in the ROM. It cannot be embedded in a chip developed by other manufacturers as the application is device specific. Considering all these issues (platform independences) Sun Microsystems Inc. in collaboration with chip manufacturer introduces Java Card Technology.

Java card is a platform on which smart card applications written using java card technology can be executed. Basically java card executes the java byte code just similar to the Java enabled web browser executes Java applets. The card operating system and application system programming can be done using java [PE01]. Java card is capable of running multiple applications so it can replace all card applications into a single card. In section 2 we discuss in brief about start card issues, software requirements and attacks.

2. Smart Card

Basically there are two types of cards, a memory card and intelligent card. A memory card can store data but cannot process it and this card can be modified or duplicated easily. A smart card is an intelligent card with a microprocessor to execute instructions on the data available in its memory resources. A smart card contains a central processing unit (CPU), Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM, Random Access Memory (RAM) and input/output unit for communication with the external device or terminal [HVMP02]. It got tamper resistant and tamper detection features inside and around the chip. Because of high security features it is used mainly to store secret information such as DES keys, RSA private-public keys or certificates for cryptography. The smart card chips are manufactured by several companies like Motorola, Philips, SGS-Thomson, Hitachi, Oki, Siemens, and Texas Instruments [GSB97].

2.1 Hardware Features

The specifications of a typical smart card are an 8-bit or 16-bit CPU, 16 to 64 Kbytes of ROM, 4 to 64 Kbytes of EEPROM and 256 bytes to 1Kbytes of RAM [HVMP02].With very limited resources available, the data storage on the smart card is managed by RAM, EEPROM and ROM. When the card is up and running RAM is used as temporary storage for loading application and data related to that application. The EEPROM stores information related to the card holder and vender like digital signature, encryption key etc., at the time of subscription. The subscriber can modify the data in EEPROM when ever required. The ROM contains the boot up programs of the smart card basically it is the operating system program [GSB97].

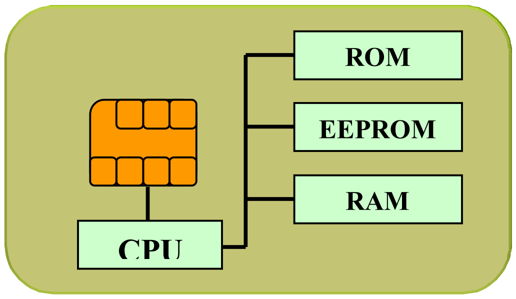

Fig 1: Smart Card

2.2 Smart Card Operating System

Card operating system is hardware specific and is embedded in ROM while manufacturing the chip. It should meet the limitation of resources on the chip and provide high level of security while handling applications and internal files of the card. Most of the present card operating systems like MULTOS are capable of running multiple applications so should provide secured environment for different vender applications [HVMP02]. Operating system has a memory management unit which guard and check the requested address before granting access permission to the data [OM99].

2.3 Smart Card Software

In the early days of smart card, applications were developed in low level languages and are embedded into card at the time of manufacturing. Low level languages are chip specific and these microcontroller programs written for one card can not be loaded on to other card. As chip manufacturers started developing their own software tools regardless of ISO 7816 standards and it became tough for programmers as the applications were platform dependent. It became big obstacle which slows down the progress of smart card implementation in the real time. This problem lead JavaSoft to developed Java Card Technology with the support of IBM, Schlumberger, Gemplus and other chip manufacturer's.

2.4 Smart Card Terminal

Smart card can communicate only through a reliable terminal. The terminal can be any thing like ATM, EFTPOS or any PC connected with a card reader. The terminal first authenticates itself and then smart card opens up for a transaction. The security issue with the terminal how trust worth are the terminals, a digital certificate can be used for secured communication between card and terminal. Basically any card application is divided into two parts, one is running on card and the other at the terminal. Most of the time both these applications are written in same application software but if the terminal need support other cards then it needs to install card terminal driver. Terminal communicates with card in the form of byte array using APDU (Application Protocol Data Unit).

2.5 Smart Card Attacks

Smart cards are vulnerable to both hardware and software attacks. Though smart card is protected by shields and layered structure there were traces of attacks where attacker reach EEPROM and read data from it. So there should be a mechanism on the chip which will detect tampering and reset all the values in memory so that secret data cannot be revealed. It also suffers software intrusion attack where attacker retrieves data by breaking the private key residing inside the smart card.

3. Java Card

Sun's JavaSoft division released their first version on Java Card 1.0 API which was implemented on Schlumberger's Cyberflex card and its later version Java Card2.0 was implemented on Gemplus GEMXpresso. Java Card specifications will help programs written in Java run on smart cards and other devices with limited processing and memory resources like mobile phone SIM. JVM occupies huge memory as it is developed keeping in view of standard computer configuration, Java Card virtual machine JCVM is reduced to fit into smart card configuration. Java card has the capability to hold multiple applications and can replace all the existing cards into a single smart card. Java card provides Open Card Framework for developing terminal applications using which platform independent terminal applications can be developed.

3.1 Java Card Features

Compared to other smart card solutions Java Card had a lot of advantages like:

- **Platform independence:** As Java Card applications comply with Java Card API specifications intend to run on all smart cards with JCRE (Java Card Runtime Environment) compatibility [HVMP02].
- **Multi-Application Support:** Java card support multi application capability which enables to load all the card applications into a single Java Card. Java card defines access restrictions over data while running multiple applications [HVMP02].
- **Power of Java:** Java Card inherits the power of Java i.e., Object Oriented Methodology and Language level security [HVMP02].

3.2 Java Card Architecture

Before going into security issues let us first discuss about Java Card Architecture. As shown in the figure-2 the low level operations like physical data transfer, file management, communication and instruction execution are handled by microcontroller program and operating system. JCRE sits over this layer and it consists of Java Card API, JCVM (Java Card Virtual Machine) and native methods. The native methods are platform dependent and they directly interact with underlying operating system, it provides services like cryptography, memory allocation and I/O operations. Java Card API also called as framework consists of all Java packages required for various Java Card functionalities. JCVM executes the Java Byte Code just like JVM but the difference is that JCVM is split into two parts. One is Java Card Converter which is off-card, converts *.class* files into card compatible format (*.cap* files) and the other is on-card part which interprets the byte code [HVMP02]. JCRE and operating system program are stored into ROM at the time of manufacture of chip while card applications are loaded into EEPROM.

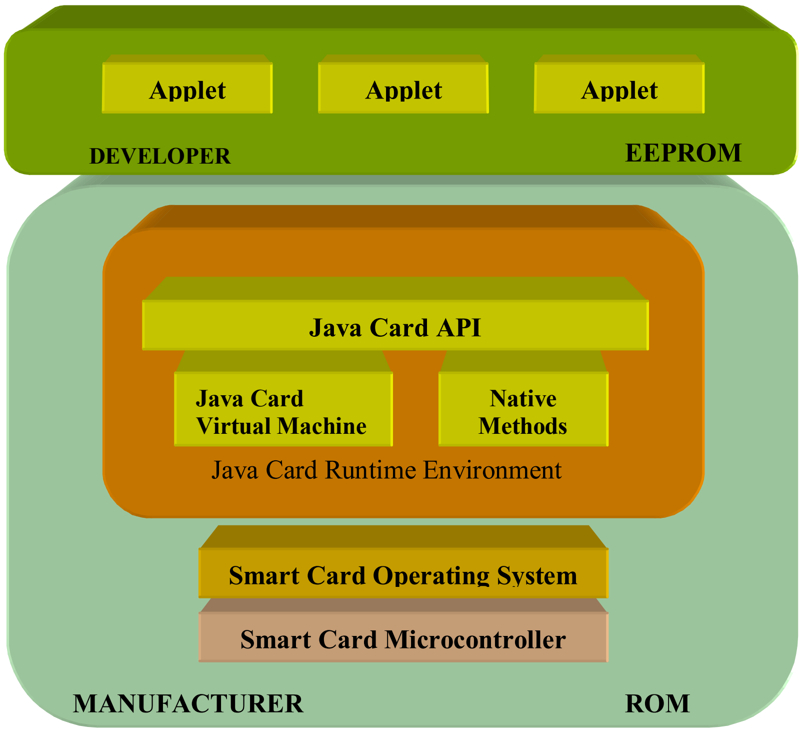
Applet
Applet
Applet
DEVELOPER
EEPROM
Java Card API
Java Card Virtual Machine
Native Methods
Java Card Runtime Environment
Smart Card Operating System
Smart Card Microcontroller
MANUFACTURER
ROM

Fig 2: Java Card Architecture [HVMP02]

3.2 Java Card Application Deployment

Java card application is first compiled using Java compiler and class files are obtained. These class files are converted into cap files (Converted Applet files) and then installed on to the card as shown in the figure 3. Now applet must be installed and registered in the card. This is done either by the installation program sitting on JCRE or by the *install* method of abstract applet class. When the installation and registration is successful, an applet is instanced and all objects related to it are created.

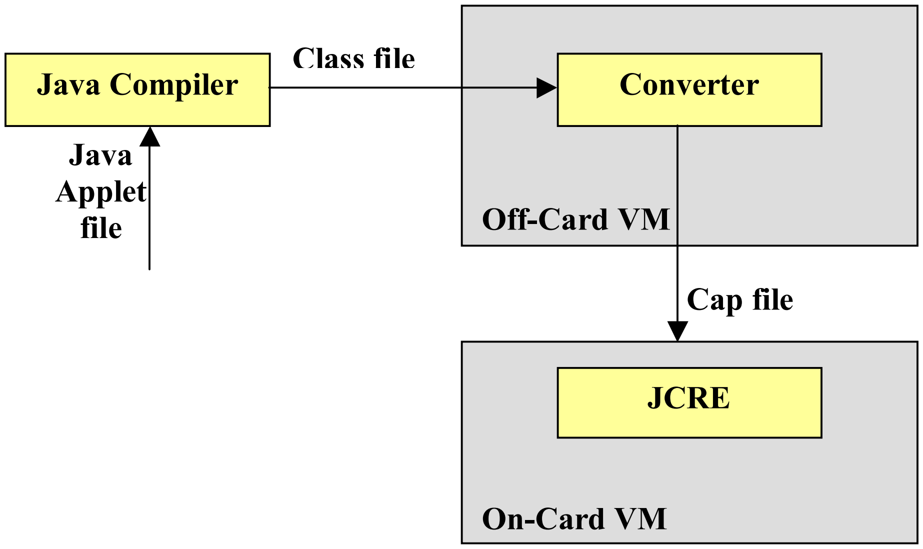

Fig 3: Java Card Applet Deployment

4. Java Card Security

Java Card provides lot of security features like applet firewalls, Java type safe, object sharing etc. As smart card uses cryptography technique for data encryption, Java Card provides Crypto API. In further sections we discuss about the features in detail.

4.1 Applet Firewall

Java Card can handle multiple applets and these applets can be from different venders so it is not secured if objects of one applet are accessible to another applet. Java Card uses applet firewall and type safe feature of Java to restrict unauthorized access of data [OM99]. That is Java Card defines context for each applet which represents all the objects accessible to a particular applet.

4.2 Object Sharing

Java Card also provides a facility to share objects by implementing *Sharable* interface. This allows the applets to share objects and at the same time it restricts them from accessing methods which are accessible only to the owner of the object that is remote method invocation is not allowed.

4.3 Java in Java Card

Java Card API and JCRE are the reduced versions of Java API and JVM. In Java Card some of the features like threads, garbage collection, security manager, dynamic class loading, cloning, finalization and real numbers computation are removed to meet the security concern and limited resource available on Card [PL01]. Dynamic class loading was always a serious threat in Java as it could load classes which JVM can not recognize till it is initiated. So a program which allocates memory continuously will lead to out of memory errors which may corrupt the card. Java Card supports packages, dynamic object loading, interfaces and exceptions etc., which adds more security to Java Card applications.

4.4 Cryptography

Java Card cryptography API specifies a set of packages that provides security features of cryptography. Cryptography is used to encrypt and hide precious data while storing and transmission. Cryptography offers following security to the data:

- **Confidentiality**-Data cannot be understood by any one other than the owner
- **Integrity-**Data can be detected if modified
- **Authentication-**Source and destination of data can be confirmed
- **Non-repudiation-**Data sender cannot deny his message

Cryptography is of two types:

Symmetric cryptography

This is a single key cryptographic technique where both the sender and the receiver should have the key. Sender encrypts data using cryptographic algorithms with a specific key and the key is provided to the receiver to decrypt the data. The symmetric cryptography can be attacked by the theft of key. But as this technique is easy and fast it is made more secured by mixing with asymmetric cryptography and used.

Asymmetric cryptography

Also called as public key cryptography, here encryption and decryption uses different keys and it is not possible to find one key from the other. All the clients are provided with a public key and private key is stored in a secured place, obviously in a smart card. Asymmetric cryptography ensures privacy of communication and it is more feasible on large networks.

Java Card uses cryptographic technique to encrypt the cap files before loading on to the card to ensure that they are trusted files [PL01]. Two packages *javacard.security and javacardx.crypto* provides interfaces for implementing cryptography on the Java Card applets and other objects.

5. Security risks in Java Card

In Java, Garbage Collection feature controls the freed and unused objects of the applet but this feature is absent in Java Card (because of its memory and processor resource limitations) which may cause serious threat like handle of a free object is allocated to a new object. If any error occurs in the program then the memory allocated to the object is never freed which lead to memory leakage.[FS99] This is a serious problem especially on smart card with such limited memory resources and application may get hanged up in the middle of transaction and suffer from denial-of-service attack.

JCRE interacts with card resources using native methods which are mostly written in C or C++, card vender can upgrade card by adding new native methods and applets can use these methods. Java card firewalls have no control over these native methods so all the firewall security fails when it comes to native methods.

Another security threat is with exception handling, it can be controlled if applets are intensively tested using testing tools before loading on card. If an exception is not handled in any of the called methods it becomes unpredictable and may hang the applet and corrupt the card.

6. Conclusion

Keeping in view of today's security concern specifically in areas like banking, health card and access control systems, smart card with Java Card will be the significant technology for many security applications. Using Java Card, smart card can be more efficient and also shifting to this technology may not be bothered by application developers as it is a familiar language and so reduces the cost of application development. As Java Card is actively under research, future versions may come with much stronger security features and compatibility. From this it is sure that Java Card will penetrate into a wide range of applications.

Reference

[PL01] PIETER H. HARTEL, LUC MOREAU. 2001. Formalizing the safety of Java, the Java virtual machine, and Java card. Pg: 517-558 ISSN: 0360-0300 ACM Press, New York, NY, USA

[OM99] Oestreicher Marcus, Transactions in Java Card, Computer Security Applications Conference, 1999. (ACSAC '99) Proceedings 15th Annual, 1999 Pg: 291 -298

[PE01] Pieter H. Hartel and Eduard de Jong, A Programming and a Modeling Perspective on the Evaluation of Java Card Implementations, LNCS 2041, pp.52-72, 2001

[FS99] Funfrocken, S Protecting mobile Web-commerce agents with smartcards Agent Systems and Applications, 1999 and Third International Symposium on Mobile Agents, Proceedings First International Symposium on, 1999 Pg: 90 -102

[HVMP02] Hassler, Vesna, Moore, Pedrick, Java card for e-payment applications, Boston, MA: Artech House, 2002. ISBN 1-58053-291-8

[GSB97] Guthery, S.B. Java card: Internet computing on a smart card Internet Computing, IEEE, Volume: 1 Issue: 1, Jan/Feb 1997 Pg: 57 -59