

# Java based Cryptographic Smart Cards (Java Cards)

Ganeshprasad Maddipati

*Abstract:* Smart Cards have been in use for the many years starting with simple bank cards & phone cards and evolved in to cards which support multiple applications. The increase in use of smart cards has increased the need for storing the information in the smart cards securely, this requirement is further increased by the presence multiple applications on the same card which means a lot of important information is present on the card. This information needs to be secured. This article is how can subset of java language called java card technology can provides for secure application development (Cards which run java based applications are called java cards). The different implementations in the real time i.e different ways of provide security for the information in the card using cryptographic algorithms . This article also discusses about the threats that are possible to java based smart card (also called java cards) and the counter measures that have been proposed for the threats specific to javacards.

***Index terms-*** Java Cards, Cryptographic applications

## I. INTRODUCTION

The smart cards have been in use for many years now. Initial cards were devices which contained information on them which was verified by the card reader and as such weren't really smart. The smart cards and the corresponding technology have evolved like every other technology from the cards which carried information on them to the cards which have a microprocessor on them and which perform operations on the information stored in the card using

the microprocessor. The cards have been developed to support multiple applications like a single card could be used as a credit card, phone card and also as identification card. Traditionally the smart card applications were developed using assembly language and this code was developed by the companies which issued the cards, so the after the application have been developed depending on the architecture of the card such applications cannot be used on cards other than ones with specific hardware and operating system. The development of multiple applications on a single card was difficult to achieve using assembly and C language and for the cards to support multi-applications the applications might come from different companies and may be used on different cards these requirements were identified by companies there was drive to develop cards and card software which can used on cards coming from different vendors. One such technology which was developed to support multiple applications was java card technology.

## Why JAVA

The SUN Microsystems developed java card technology which was a subset of its java programming language the reason for develop the technology as the java language in general these advantages over other programming languages in development of multiple applications

- 1) Java is platform independent and so the code developed is independent of the underlying hardware which varies for each card vendor.
- 2) Java being object oriented language ,the application development is easier.
- 3) Java has inherent security features which can utilized to develop secured applications.

## II. JAVA CARD TECHNOLOGY

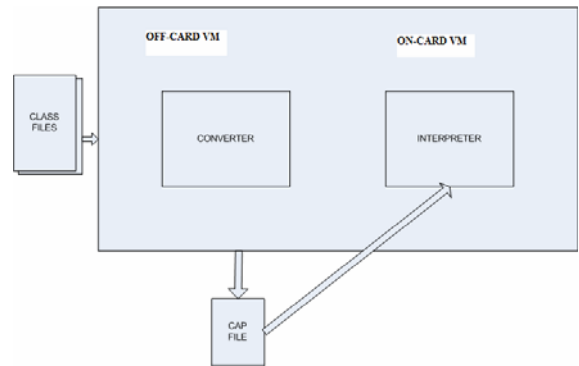
The SUN Microsystems developed the java card technology for supporting the application development as the smart cards have constrained resources in regard to memory and processing capabilities the actual java cannot be used to program the application ,so the Java card technology uses the important features of the java language and leaves out features are not required for smart card and also the hard ware that is required for java cards is different from the normal smart cards as this technology was mainly developed for multiple applications to be present on the same card. The reason for using the subset of the actual language was based on memory constrains of the cards and also for application security.

part one part lies in the card acceptance device(CAD) and the other lies on the card and also the JCVM is also subset of the actual java virtual machine.

The javacard applet written is first compiled in a java compiler which produces a class file which contains byte code and is portable, The class file is debugged using debuggers like javacard workstation development environment (JCWDE) which is provide by the java card technology tool kit.which simulates the on card virtual machine.

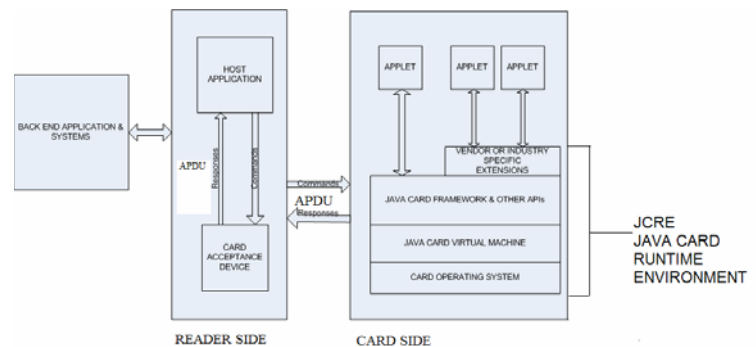
The class file is then converted into an CAP(Converted applet )file using the converter which is part of the off card virtual machine and also includes a verifier which verifies if the applet uses the correct API supported by the java card technology.

Supported Java Features	Unsupported Java Features
Small primitive data types: boolean,byte,short	Large primitive data types: long ,double ,float
One dimensional arrays	characters and strings
Java packages, classes interfaces and exceptions	Multidimensional arrays
Java object oriented features: Inheritance, virtual methods, overloading and dynamic object creation, access scope and binding rules	Dynamic class loading
The int keyword and 32 bit integer data type support are optional	Security Manager
	Garbage Collection and finalization
	Threads
	Object Serialization
	Object cloning



The APDU( Application Protocol Data Unit) is part of the ISO 7816 standard which is used to as unit for communication between the host application and the card.

The whole model of java card technology would be as shown in the figure below



**How does a Java card Technology work?**

The java card technology has the following parts for the creation of java card applets(applications for the card).

- The java card API specification
- The java card Run Time(JCRE)specification
- The java card virtual machine(JCVM) specification.

The java card virtual machine due to the system constrains and also for security reasons is divided two

### III. Java Cryptographic API

The java card technology includes classes and API's which have been developed specifically for the java cards like the javacard framework API, Javacardx crypto api which has all algorithms like RSA, DES, DSA, SHA-1MD5 defined in them for usage in development of secure applications.

The java card technology has been developed to be used with the existing smart cards standard (ISO 7816).

The java card technology provides the following cryptographic API's which can be used in the secure application development.

Package Name	Description
javacard.framework	This is the core package on the card. It defines classes such as Applet and PIN, which are the fundamental building blocks for Java Card programs and APDU, System and Util, which provide runtime and system service to Java Card programs, such as APDU handling and object sharing.
javacardx.framework	This package provides an object-oriented design for an ISO 7816-4 compatible file system.
javacard.security, javacardx.crypto & javacardx.cryptoEnc	Those two packages support cryptographic functionality required in smart cards.

The cryptographic APIs have been designed around these principles

- Implementation independence and interoperability
- Algorithm independence and extensibility

The Crypto APIs are structured into two packages: javacard.security and javacardx.crypto. The javacard.security package consists of various interfaces for implementing symmetric and

asymmetric keys, the key builder class, the authentication class (message digest and signature) and the class for generating random data.

The package javacardx.crypto is an extension and contains the cipher class which allows the use of strong encryption and the KeyEncryption interface for enabling a key implementation to access the encrypted key data.

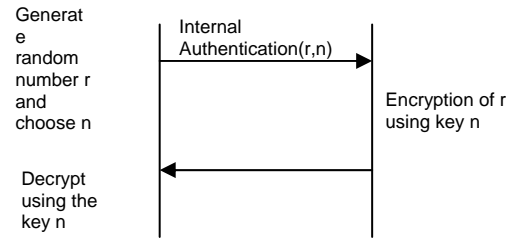
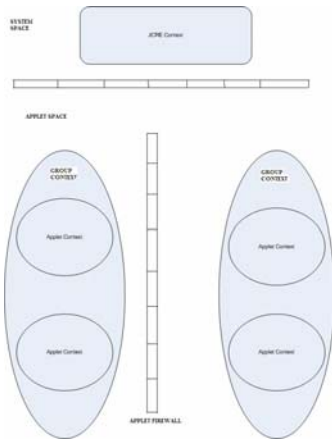
The java card crypto API provides to sign data using RSA with Chinese remainder theorem and without Chinese remainder theorem. The most of the classes in the crypto API are abstract classes and need to be extended using implementation class. The RandomData class can be used to create random numbers for cryptographic operations.

The security features provided by java which are present in the java card technology are

- 1) java language is strongly typed
- 2) java language enforces boundary checks on array access
- 3) java language has no pointer arithmetic.
- 4) variables need to be initialized before they can be used.
- 5) The level of access for methods and classes can be controlled.

The additional features that are not provided by java language and are present in the java card technology are

- 1) Transient and persistent object models: It means the objects are stored in persistent memory and the temporary data is stored in transient objects in RAM which can be programmed to be cleared on reset or deselected.
- 2) Atomicity and transactions: The data stored in persistent memory when ever there is an update to the data, been done when ever there is a failure in the update, all the changed values are reset to the original values even those which have successful update during that instance.
- 3) The javacard.framework.Util tries to implement the above feature in the array of values.
- 4) Applet Firewall: The security and integrity of each applet residing on a java card are protected by the applet firewall. The applet firewall enforces applet isolation and separates the system space from applet space.



r is the random number generated by external entity and n is the key number

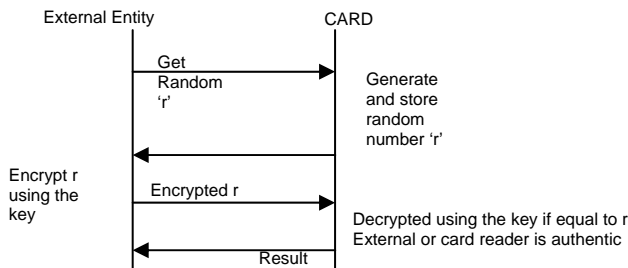
5)Object sharing: In java cards system it is implemented by the java card run time environment which make decisions regarding the object sharing and it has highest privilege. The JCRE uses the objects called shareable interface objects to implement the object sharing.

6)Native Methods;The native methods are not executed by the JAVA card virtual machine and so are not subject to security protections of java card platform only the applications which reside in the ROM are allowed to use native methods.

#### IV. Cryptographic Protocols

These are the protocols devised to ensure the security of the card by identification of the card and the external entity and also providing for secure communication between the card and the external entity

##### 1)External Authentication



##### 2)Internal Authentication

#### 3)Secure Messaging

- I)Protected mode operation
- II)Encrypted and Protected mode operation
- III)Key derivation using Session keys

All the cryptographic protocols use either the public key or symmetric key schemes. Public key schemes like RSA and symmetric key scheme like DES are the most popular algorithms.

#### V. Security Threats for Java Cards

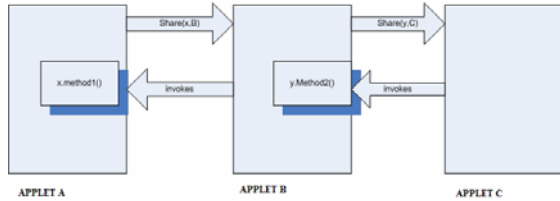
The security attacks that are possible against smart cards like the physical attacks and side channel attacks are still possible on the java cards and the java card technology does not provide any features to prevent them. In addition to these attacks there are few more security threats which are possible and are listed below.

- 1) *Introduction of Malicious applet(code) on to the card this threat can be used to read the data in other classes especially crypto applets so that they access the keys or make the card fail by overflowing the card memory by creating exceptions which cannot be caught.*
- 2) *Exploitation of bugs in java language by applet deployed on the card*
- 3) *Exploitation of applet which has not been developed properly.*

These threats can be countered by verifying the applet that is being loaded on to the card by using some verification process and also the applet needs to be developed securely and verified .The CAP file verification by using signature generation i.e signing of the CAP files and also verifying the java card run time environment for bugs in it implementation.

The share interface object also introduces threats which can be summarized from the figure below.

Suppose the applet A is an cryptographic applet and provide security for the card and its method x contains the key generation method which is shared by applet B and there is some method y which is shared by applet C since the method X is accessible to the method y of the applet B which in turn is accessible to applet C this mean the applet C can use method y to access the method x of applet A which is other wise unaccessible to C directly and since this method stores the key information this can be accessed by 'C' which is a security threat.



**VI. Model 330 Java Card**

The model 330 java card is a type of java card which is different from the normal java card. This card was developed by DATAKEY Corp and conforms with FIPS 140-2 standards it provides an applet called Java Card Cryptographic Operating System (JCCOS) applet which resides on the card operating system and on which the other applets are developed this JCCOS applet has been designed so as to control the sharing of the objects between applets. It also implements all the cryptographic operations and provides its services to be used by the applet present on it.

The JCCOS applet has two roles one is the Security officer role and other is User role

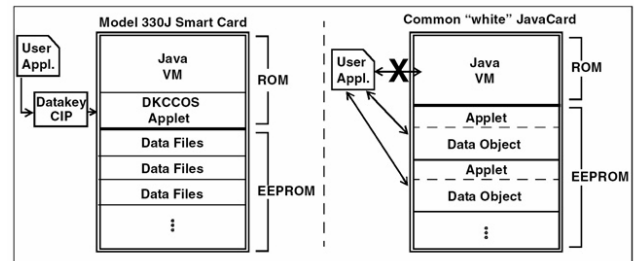
The JCCOS Security Officer (SO) role is responsible for

- configuring the applet by changing the configuration file
- settings (specifying which algorithms are allowed by the applet, which keys may be generated, and who may generate keys), setting up the User's password and
- unblock User PINs.

The User role is essentially the end user and thus has access to all of the cryptographic functions of the module, but does not have the access (that the

Security Officer has) to the card configuration functions.

The difference between the java card and model 330 java card (produced by Safe Net) can be summarized from the figure below



**VII. Commercially available Cards based on java card technology :**

	Sm@rtCafé	GemXpresso 211	Cyberflex Access	Model 330J(using JCCOS)
Manufacturer	Giesecke&Devri ent	Gemplus	Schlumberger	SafeNet
Resources	1,280 bytes RAM, 32KB ROM ,16KB EEPROM	2KB RAM,32 KB ROM,32 KB EEPROM	16KB EEPROM	2300 bytes RAM,96KB ROM,32KB EEPROM
Protocols supported	T=0,T=1	T=0,T=1	T=0	T=1
Java Card Version	Java Card 2.1	Java Card 2.1	Java Card 2.0	Java Card 2.1
Cryptographic algorithms supported	DES, Triple DES, RSA, SHA-1	DES, Triple DES	DES, Triple DES, RSA, SHA-1	DES, Triple DES, RSA, SHA-1
Security Services	External & Mutual authentication, digital Signature, session key generation	-	External & Internal authentication	External and Internal authentication Protection against DPA & timing analysis

**VIII. Conclusion**

The java card technology is still evolving and the also there are different implementations being developed as such as JCCOS for secure java cards and also there a tough competition for java card technology from the MULTOS technology which also provides for multi-application cards and also provide good amount of security.

The future depends which card can provide more security at lesser cost and also support any newer applications that come up in the future.

#### LIST OF REFERENCES:

1)Chapters(3,4,9,10,11) "JAVA CARD Technology for Smart Cards" architecture and programmer's guide by Zhiquen Chen

2)Smart Card Application Development using Java by Uwe Hansmann

3)"Java Card for E-payment Applications" by Vesna Hassler,Martin Manninger

4)"JAVA CARD SECURITY" [www.riscure.com](http://www.riscure.com)

5)"MODEL 330J smart card"

[http://www.hmk.de/downloads/datakey/Model\\_330J\\_Smart\\_Card.pdf](http://www.hmk.de/downloads/datakey/Model_330J_Smart_Card.pdf)