

ECC BASED PASSWORD AUTHENTICATED KEY AGREEMENT SCHEME FOR SMART CARDS

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

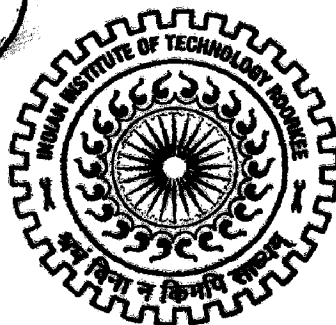
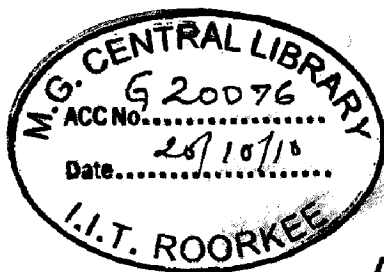
MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

AQEEL KHALIQUE



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)**

JUNE, 2010

Candidate Declaration

I hereby declare that the work being presented in the dissertation report titled “**ECC Based Password Authenticated Key Agreement Scheme For Smart Cards**” in partial fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science and Engineering, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, is an authentic record of my own work carried out under the guidance of Dr Kuldip Singh, in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee. I have not submitted the matter embodied in this dissertation report for the award of any other degree.

Dated: 02-06-10

Place: IIT Roorkee.

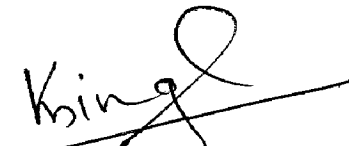

(Aqeel Khalique)

Certificate

This is to certify that above statements made by the candidate are correct to the best of my knowledge and belief.

Dated: 02-06-10

Place: IIT Roorkee.


Dr. Kuldip Singh,
Professor,
Department of Electronics
and Computer Engineering,
IIT Roorkee, Roorkee,
247667 (India)

ABSTRACT

In the ever expanding digital world, cryptography is becoming more and more important to provide services such as encryption, digital signatures and key establishment. One of the services provided by cryptography is authentication. Authentication is used to prove that the user is the desired entity for the particular communication. Nowadays, human had come out of this physical restriction for communication and transaction. One need to be authenticated and can fulfill his desired task, whether it is shopping, insurance, availing medical facilities, monetary transaction etc. One of the modes of being authenticated is to use smart cards. Smart cards are widely in used and are thus prone to attacks, such as theft, misuse and unauthenticated usage. There are various authentication schemes which are being used to provide adequate security measures apart from authentication itself. Some of them involve various cryptographic methods including public key cryptography. These schemes are measured on parameters of computational efficiency and strength to withstand the privacy of secret key. One of the emerging areas is elliptic curve cryptosystem, which has greater strength at comparatively smaller key lengths of other cryptographic schemes based on integer factorization and discrete logarithms.

ECC is based on elliptic curves defined over a finite field; binary or prime. ECC is used in constrained devices which have low memory, low processing power and low computation power. Thus smart cards are best suited to imply ECC to provide authentication.

In this dissertation, an ECC based scheme is proposed to authenticate smart cards. The scheme is quite effective and simple and also provides desired security features. The scheme is implemented on Java Card Kit. Lastly, the cost and functionality analysis of the scheme is done and various security aspects have been addressed.

Table of Contents

CANDIDATE'S DECLARATION	i
CERTIFICATE	i
ACKNOWLEDGMENTS	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ACRONYMS	ix
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Organization of the Report	3
Chapter 2: Preliminaries	5
2.1 Overview of Cryptography	5
2.1.1 Integer Factorization	6
2.1.2 Discrete Logarithm	7
2.1.2.1 Comparison with Integer Factorization	8
2.1.3 Elliptic Curve Discrete Logarithm	8
2.2 Elliptic Curve Cryptography	9
2.2.1 Mathematical Background on Elliptic Curves	11
2.2.2 Elliptic Curve over Finite Field Z_p	12
2.3 Finite Fields	14
2.3.1 Prime Field F_p	14
2.3.1.1 Algebraic Formulae over F_p	15
2.3.2 Binary Field F_2^m	16
2.3.2.1 Algebraic Formulae over F_2^m	17
2.4 Elliptic Curves Operations over Finite Fields	18

2.4.1 Point Addition	18
2.4.2 Point Doubling	19
Chapter 3: Smart Cards	21
3.1 History	21
3.2 Types of Smart Cards	22
3.2.1 Memory Chip Cards	23
3.2.2 Microprocessor Cards (Smart Cards)	24
3.2.3 Contact Based Smart Cards	24
3.2.4 Contactless Smart Cards	25
3.3 Limitations of Smart Cards	26
3.4 Meeting the Implementation Constraints with ECC	27
Chapter 4: Mutual Authentication	29
4.1 Overview of Authentication Schemes	29
4.2 Authenticated Key Agreement.....	30
4.3 Key Establishment Protocol	31
4.3.1 Implicit Key Authentication	32
4.3.2 Explicit Key Authentication	32
4.4 Proposed Scheme based on Authenticated Key Agreement	32
Chapter 5: Implementation Details	35
5.1 Generation of Domain Parameters over Elliptic Curve	35
5.2 Java Implementation	35
5.2.1 RMI	35
5.2.2 Java Card Kit	36
5.2.3 Package Structure and Classes.....	37
Chapter 6: Security Analysis and Comparison	39
6.1 Security Analysis	39
6.2 Comparison with Related Scheme	40
Chapter 7: Conclusions and Future Work.....	45
7.1 Conclusion	45
7.2 Future Work	45

REFERENCES	47
PUBLICATIONS	50
APPENDIX	
1. Domain Parameters by NIST	51

List of Figures

- Figure 2.1: Example of elliptic curve: $y^2=x^3-x$
- Figure 2.2: Example of elliptic curve: $y^2=x^3+x+1$
- Figure 2.3: The Elliptic Curve $E_{23}(1,1)$
- Figure 2.4: The Elliptic Curve $E_{2^4}(g^4, 1)$
- Figure 2.5: Point Addition
- Figure 2.6: Point Doubling
- Figure 3.1: Worldwide Growth of Smart Cards
- Figure 3.2: Types of Smart Cards
- Figure 3.3: Contact Based Smart Cards
- Figure 3.4: Contactless Smart Cards
- Figure 3.5: Relative Factor Chip Area
- Figure 4.1: Authentication Phase of the Proposed Scheme
- Figure 5.1: Installer Components of Java Card Technology
- Figure 6.1: Comparison of Related scheme with Proposed scheme
- Figure 6.2: Comparative Graph of Computation Complexities

List of Tables

Table 2.1: Equivalent Security at different Key Lengths

Table 2.2: Points on the Elliptic Curve $E_{23}(1,1)$

Table 2.3: Points on the Elliptic Curve $E_{2^4}(g^4, 1)$

Table 6.1: Functionality

Table 6.2: Storage Cost

Table 6.3: Communication Cost

Table 6.4: Computation Cost of Juang *et al.* scheme

Table 6.5: Computation Cost of Fan *et al.* scheme

Table 6.6: Computation Cost of Proposed scheme

List of Acronyms

Acronym	Full Form
AKA	Authenticated Key Agreement
AKC	Agreement with Key Confirmation
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
AS	Authentication Server
CAP	Converted Applet
CDHP	Computational Diffie-Hellman Problem
DH	Diffie-Hellman
DL	Discrete Logarithm
DLP	Discrete Logarithm Problem
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECFP	Elliptic Curve Factorization Problem
EEPROM	Electrically Erasable Programmable Read-Only Memory
EKA	Explicit Key Authentication

FIPS	Federal Information Processing Standards
GSM	Global System for Mobile Communications
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IKA	Implicit Key Authentication
ISO	International Organization for Standardization
JCWDE	Java Card Workstation Development Environment
KAC	Key Authentication Center
NIST	National Institute of Standards and Technology
PDA	Personal Digital Assistant
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PW	PassWord
RAM	Random Access Memory
ROM	Read Only Memory
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
XOR	eXclusive OR

1.1 Introduction

With the need for information security in today's digital systems both acute and growing, cryptography has become one of their critical components. Cryptographic services are required across a variety of platforms in a wide range of applications such as secure access to private networks, stored value, electronic commerce, and health care. Incorporating these services into solutions presents an ongoing challenge to manufacturers, systems integrators and service providers because applications must meet the market requirements of mobility, performance, convenience, and cost containment. With the advancement and tremendous development of wireless technology, various mobile devices have prevailed in our daily life, e.g. cell phone, PDA, smart cards and so on. By using these mobile devices, people can accomplish the electronic transactions anytime and anywhere. Thus, more and more electronic transactions for mobile devices are implemented on Internet or wireless networks. With the rapid growth of industrial network technology [1, 2, 3, 4, 5, 6], user authentication takes an important role for achieving the dependable network environments. The goal of user authentication is to provide the communicating entities with some assurance that they know each other's true identities. There is the additional goal that the two entities end up sharing a common key called a session key known only to them. This session key can then be used for some time thereafter to provide privacy, data integrity, or both. There is a vast literature on authenticated key agreement schemes. We refer the reader to [7] for a more extensive historical discussion and to [8] and [9] for formal model approaches to design and analyze authenticated key agreement schemes.

To access resources from a remote system, users should have proper access rights. One of the simple and more efficient mechanisms is the use of a password authentication scheme. To access the resources, each user should have an identity (ID) and a password (PW). In the existing traditional setup the ID and PW are maintained by the remote system in a verification table. If a user wants to log in a remote server, he has to submit his ID and PW to the server. The remote server

receives the login message and checks the eligibility of the user by referencing the password or verification table. If the submitted ID and PW match the corresponding pair stored in the server's verification table, the user will be granted access to the server.

A remote password authentication scheme is used to authenticate the legitimacy of the remote user over an insecure channel. In such a scheme, the password is often regarded as a secret shared between the authentication server (AS) and the user, and serves to authenticate the identity of the individual login. Through knowledge of the password, the remote user can create a valid login message to the authentication server. AS check the validity of the login message to provide the access right. Password authentication schemes with smart cards have a long history in the remote user authentication environment.

Two problems are found in this existing traditional mechanism.

1. The administrator of the server will come to know the password, because the server maintains the password table.
2. An intruder can impersonate a legal user by stealing the user's ID and PW from the password table.

To add to the woes, the current Internet is vulnerable to various attacks such as denial of service attack, forgery attack, forward secrecy attack, server spoofing attack, parallel session attack, guessing attack, replay attack, smart card loss attack, and stolen verifier attack.

1.2 Motivation

The main motivation behind this scheme is the security of elliptic curve cryptography (ECC). ECC is fully exponential problem based on Elliptic Curve Discrete Logarithmic Problem (ECDLP). ECDLP is the inverse operation to point multiplication. In ECC, the elliptic curve is used to define the members of the set over which the group is calculated, as well as the operations between them which define how mathematics works in the group [10].

ECC offers considerably greater security for a given key size. The smaller key size also makes possible much more compact implementations for a given level of security, which means faster cryptographic operations, running on smaller chips or more compact software. This results in less heat production and less power consumption which is of particular advantage in constrained devices.

Today the communication over Internet is vulnerable to various attacks such as denial of service attack, forgery attack, forward secrecy attack, server spoofing attack, parallel session attack, guessing attack, replay attack, offline dictionary attack, smart card loss attack, and stolen verifier attack. So the aim is to provide a scheme which can withstand all these attacks and efficient in terms of computational, communicational and storage cost.

1.3 Problem Statement

In this dissertation work we propose and implemented an ECC based password-authenticated key agreement scheme. Various security parameters are addressed and functionalities have been observed and compared.

1.4 Organization of the Report

This dissertation report proposes a password-authenticated key agreement scheme based on ECC. The scheme is implemented on Java Card Kit to emulate the authentication as used in smart cards to provide better security in constrained environment. The organization of the report is as follows:

Chapter 2 discusses the background of ECC and other cryptographic schemes, and mathematical details of ECC concerning the finite fields and elliptic curve operations over elliptic curves.

Chapter 3 discusses the smart cards and their suitability to use ECC for greater security and their competency as constrained devices.

Chapter 4 describes proposed scheme and also describes few aspects of desired security features such as explicit key confirmation and authenticated key agreement.

Chapter 5 presents a brief description of the implementation of the proposed scheme and also draw some focus towards the Java Card Technology. It covers the details of how to generate the CAP file which is used in smart cards to execute the commands in the card to communicate with the server.

Chapter 6 discusses how the proposed scheme satisfies the security issues. It then presents the results of the scheme and then shows a brief but important comparison with related scheme.

Chapter 7 concludes the dissertation work and gives suggestions for future work.

2.1 Overview of Cryptography

The proliferation of computers and communications systems in the 1960s brought with it a demand from the private sector for means to protect information in digital form and to provide security services. Over the centuries, an elaborate set of protocols and mechanisms has been created to deal with information security issues when the information is conveyed by physical documents. Often the objectives of information security cannot solely be achieved through mathematical algorithms and protocols alone, but require procedural techniques and abidance of laws to achieve the desired result. Cryptography is the branch of cryptology dealing with the design of algorithms for encryption and decryption, intended to ensure the secrecy and/or authenticity of message. These algorithms can be categorized into two category namely, symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography a single key, known as private key is shared between the communicating parties only. Both encryption and decryption is done using this shared secret key. As number of communicating parties increased the management of these keys become a serious issue as it required $N*(N-1)/2$ keys, where N is the number of communicating entities. In asymmetric key cryptography, known as public key cryptography, there is a pair of keys known as private key (known only to the entity) and public key (known to all others). Here, the key management is not such an issue as it required $2N$ keys in N communicating entities.

Among the services provided by the cryptographic schemes, the following four form a framework upon which the others will be derived: (1) privacy or confidentiality; (2) data integrity; (3) authentication; and (4) non-repudiation.

1. Confidentiality is a service used to keep the content of information from all but those authorized to have it. Secrecy is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

2. Data integrity is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by

unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

3. Authentication is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes: entity authentication and data origin authentication. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).

4. Non-repudiation is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute.

A fundamental goal of cryptography is to adequately address these four areas in both theory and practice.

The cryptographic schemes are generally based on Integer Factorization, Discrete Logarithm, and Elliptic Curve Cryptography (ECC). ECC was first introduced by Koblitz [10] and Miller [11] is the best cryptographic system known so far as it provides equivalent level of security as of RSA, DSA at smaller key lengths. More details on ECC will be discussed later. The former two schemes have been briefly stated to have an idea of their functionality.

2.1.1 Integer Factorization

In Integer factorization, given an integer n which is the product of two large primes p and q such that:

$$n = p * q \tag{2.1}$$

It is easy to calculate n for given p and q but it is computationally infeasible to determine p and q given n for large values of n . One of the famous algorithms is RSA.

Its security depends on the difficulty of factoring the large prime numbers. The best known method for solving Integer Factorization problem is Number Field Sieve which is a sub-exponential algorithm and having a running time of $\exp[1.923*(\log n)^{1/3}*(\log \log n)^{2/3}]$ [12].

2.1.2 Discrete Logarithm

Discrete logarithms are ordinary logarithms involving group theory. An ordinary logarithm $\log_a(b)$ is a solution of the equation $a^x = b$ over the real or complex numbers. Similarly, if g and h are elements of a finite cyclic group G then a solution x of the equation $g^x = h$ is called a discrete logarithm to the base g of h in the group G , i.e. $\log_g(h)$

A group with an operation $*$ is defined on pairs of elements of G . The operations satisfy the following properties:

1. **Closure:** $a * b \in G$ for all $a, b \in G$.
2. **Associativity:** $a * (b * c) = (a * b) * c$ for all $a, b \in G$.
3. **Existence of Identity:** There exists an element $e \in G$, called the identity, such that $e * a = a * e = a$ for all $a \in G$.
4. **Existence of inverse:** For each $a \in G$ there is an element $b \in G$ such that $a * b = b * a = e$. The element b is called the inverse of a .

Moreover, a group G is said to be abelian if $a * b = b * a$ for all $a, b \in G$. The order of a group G is the number of elements in G .

The discrete logarithm problem is to find an integer x , $0 \leq x \leq n-1$, such that $g^x \equiv h \pmod{p}$, for given $g \in \mathbb{Z}^*_p$ of order n and given $h \in \mathbb{Z}^*_p$. The integer x is called the discrete logarithm of h to the base g .

Digital Signature Algorithm (DSA), Diffie Hellman (DH) and El Gamal are based on discrete logarithms.

The best known method for solving Discrete Logarithm problem is Number Field Sieve which is a sub-exponential algorithm, having a running time of $\exp[1.923*(\log n)^{1/3}*(\log \log n)^{2/3}]$ [12].

2.1.2.1 Comparison with Integer Factorization

While the problem of computing discrete logarithms and the problem of integer factorization are distinct problems they share some properties:

- both problems are difficult (no efficient algorithms are known for non-quantum computers),
- for both problems efficient algorithms on quantum computers are known,
- algorithms from one problem are often adapted to the other, and
- the difficulty of both problems has been exploited to construct various cryptographic (code) systems.

2.1.3 Elliptic Curve Discrete Logarithm

An elliptic curve E_k , [10] defined over a field κ of characteristic $\neq 2$ or 3 is the set of solutions $(x, y) \in \kappa^2$ to the equation

$$y^2 = x^3 + ax + b \quad (2.2)$$

$a, b \in \kappa$ (where the cubic on the right has no multiple roots).

Two nonnegative integers, a and b , less than p that satisfy:

$$4a^3 + 27b^2 \pmod{p} \neq 0 \quad (2.3)$$

Then $E_p(a, b)$ denotes the elliptic group mod p whose elements (x, y) are pairs of nonnegative integers less than p satisfying:

$$y^2 = x^3 + ax + b \pmod{p} \quad (2.4)$$

together with the point at infinity O .

The elliptic curve discrete logarithm problem can be stated as follows. Fix a prime p and an elliptic curve.

$$Q = xP \quad (2.5)$$

where xP represents the point P on elliptic curve added to itself x times. Then the elliptic curve discrete logarithm problem is to determine x given P and Q . It is relatively easy to calculate Q given x and P , but it is very hard to determine x given Q and P .

ECC is based on ECDLP. ECDH and ECDSA are cryptographic schemes based on ECC. The best known algorithm for solving ECDLP is Pollard-Rho algorithm which is fully exponential having a running time of $\sqrt{\left(\frac{\pi n}{2}\right)}$ [12].

2.2. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) can be regarded as elliptic curve analogues of the traditional discrete logarithm (DL) cryptosystems in which a finite cyclic group is replaced by the group of points on an elliptic curve over a finite field. The security of ECC depends on the computational intractability of the elliptic curve discrete logarithms problem (ECDLP) which is significantly harder than the Discrete Logarithm Problem (DLP), that is, smaller keys can be used in ECC than with DL but with equivalent levels of security. Comparing with other asymmetric cryptography, ECC can provide stronger security and faster computations using shorter key length, which make it the most suitable for wireless and mobile communication systems, including smartcards and handheld devices which are constrained related to processor speed, bandwidth, security and memory. Table 2.1 shows the relative comparison between integer factorization, discrete logarithm and elliptic curve discrete logarithm based techniques on the basis of key length providing equivalent security.

Table 2.1: Equivalent Security at different Key Lengths

Symmetric(in bits)	RSA/DSA/DH (in bits)	ECC (in bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

In 1985, Koblitz [10] and Miller [11] independently proposed using the group of points on an elliptic curve defined over a finite field in discrete log cryptosystems. It is known that the cyclic subgroup of points on an elliptic curve would have numerous advantages over the multiplicative group of finite fields, used in the Diffie-Hellman scheme and other algorithms used today. First, the discrete logarithm problem (DLP) in the group of elliptic curves is computationally more complex than that of taking the discrete logarithm in finite fields. And second, the structure of the point group on the

elliptic curve will admit the parameters of the cryptosystem to be chosen more flexibly. Elliptic curves over finite fields provide an exhaustible supply of finite abelian groups that are obedient to computation because of their prolific structure. In several ways, elliptic curves are natural analogs to the multiplicative group of fields, and at the same time have the advantage that it has more adaptability in choosing an elliptic curve than in choosing a finite field. Elliptic Curves provide constructing elements with certain rules, which define its group operation. These groups don't have certain properties that may facilitate the most powerful cryptanalysis. The primary advantage that elliptic curve systems have over systems based on the multiplicative group of a finite field is the absence of a sub exponential-time algorithm that could find discrete logs in these groups. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security.

The results are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards, PC (personal computer) cards, and wireless devices. Generally, the security of ECC relies on the difficulties of the following problems [13].

- **Definition 1** Given two points P and Q over $E_p(a, b)$, the elliptic curve discrete logarithm problem (ECDLP) is to find an integer $s \in F_p^*$ such that $Q = s.P$.
- **Definition 2** Given three points P , $s.P$, and $t.P$ over $E_p(a, b)$ for $s, t \in F_p^*$, the computational Diffie-Hellman problem (CDHP) is to find the point $(s.t).P$ over $E_p(a, b)$.
- **Definition 3** Given two points P and $Q = s.P + t.P$ over $E_p(a, b)$ for $s, t \in F_p^*$, the elliptic curve factorization problem (ECFP) is to find two points $s.P$ and $t.P$ over $E_p(a, b)$.

Up to now, there is no algorithm to be able to solve any of the above problems [13]. After two decades of research and development, elliptic curve cryptography (ECC) has now gained widespread exposure and acceptance, and has ultimately moved from being an interesting mathematical construction to a well-established public-key cryptosystem already included in numerous standards and adopted by an increasing

number of companies. All elliptic curve cryptosystems are based on an operation called elliptic curve point multiplication which is defined as

$$Q = kP = \underbrace{P + P + \dots + P}_k \quad (2.6)$$

where k is an integer and Q and P are points on an elliptic curve. A point is represented with two coordinates as (x, y) . The reason why elliptic curve point multiplication is used in cryptosystem is that it is relatively easy to compute but its inverse operation called elliptic curve discrete logarithm problem, that is finding k if P and Q are known, is considered impossible to solve with present computational resources if parameters are chosen correctly.

ECC is now accepted worldwide by NIST, FIPS 186, ANSI and IEEE P1363, FIPS 140-2# X9.62 – Elliptic Curve Digital Signature Algorithm (ECDSA), # X9.63 – Key Agreement and Key Transport Using Elliptic Curve Cryptography.

2.2.1 Mathematical Background on Elliptic Curves

The name elliptic curves come from the observation that these curves arise in studying the problem of how to compute the arc length of an ellipse. If one writes down the integral, which gives the arc length of an ellipse and makes an elementary substitution, the integrand will involve the square root of a cubic or quadratic polynomial. To compute the arc length of an ellipse, one integrates a function involving

$$y = \sqrt{f(x)} \quad (2.7)$$

and the answer is given in terms of certain functions on the elliptic curve

$$y^2 = f(x) \quad (2.8)$$

Elliptic curves exist over any field (e.g. real, complex, finite), but for cryptographic purposes we will only be concerned with those over finite fields.

Let assume K is a field with characteristic $\text{char}(K)$. K can be either the field R of real numbers, the field C of complex numbers, or the finite field $GF(q)$ with $q=px$ elements and p prime. An elliptic curve over K , denoted by E , is a curve of the first kind in the set of points (x, y, z) , which satisfy the (*Weierstrass*) equation:

$$E: y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz + a_5 \quad (2.9)$$

with integral coefficients $a_i \in K'$ (K' is a fixed algebraic closure of K).

There is exactly one point in E with z -coordinate equal to 0, that is $(0,1,0)$. We call this point the point at infinity and denote it by O . To obtain an equation in *affine coordinate*, we will write *Weierstrass* equation for an elliptic curve using non-homogeneous (*affine*) coordinate $x = X/Z, y = Y/Z$, and together with the extra point at infinity O . We get,

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2.10)$$

Elliptic curves can be simplified over fields of different characteristic $\text{char}(K)$ and can be examined by means of coordinate transformations. We distinguish between the following three cases:

- $\text{char}(K) = 2$:

$$E: y^2 + xy = x^3 + ax^2 + c \quad (2.11)$$

$$E: y^2 + ay = x^3 + bx + c \quad (2.12)$$

- $\text{char}(K) = 3$:

$$E: y^2 = x^3 + ax^2 + bx + c \quad (2.13)$$

- $\text{char}(K) > 3$:

$$E: y^2 = x^3 + ax + b \quad (2.14)$$

2.2.2 Elliptic Curve over Finite Field Z_p

An elliptic curve E over Z_p is defined in the Cartesian coordinate system by an equation of the form:

$$y^2 = x^3 + ax + b \quad (2.15)$$

where $a, b \in Z_p$, and

$$4a^3 + 27b^2 \neq 0 \pmod{p}, \quad (2.16)$$

together with a special point O , called the point at infinity. The set $E(Z_p)$ consists of all points $(x; y)$, $x \in Z_p, y \in Z_p$, which satisfy the defining equation, together with O . An elliptic curve is defined by an equation in two variables with coefficients, which are restricted to elements in a finite field, resulting in the definition of a finite abelian group. Figure 2.1 and 2.2 shows an elliptic curve. For given values of a and b , the plot consists of negative and positive values of y for each value of x . Thus each curve is symmetric about $y=0$.

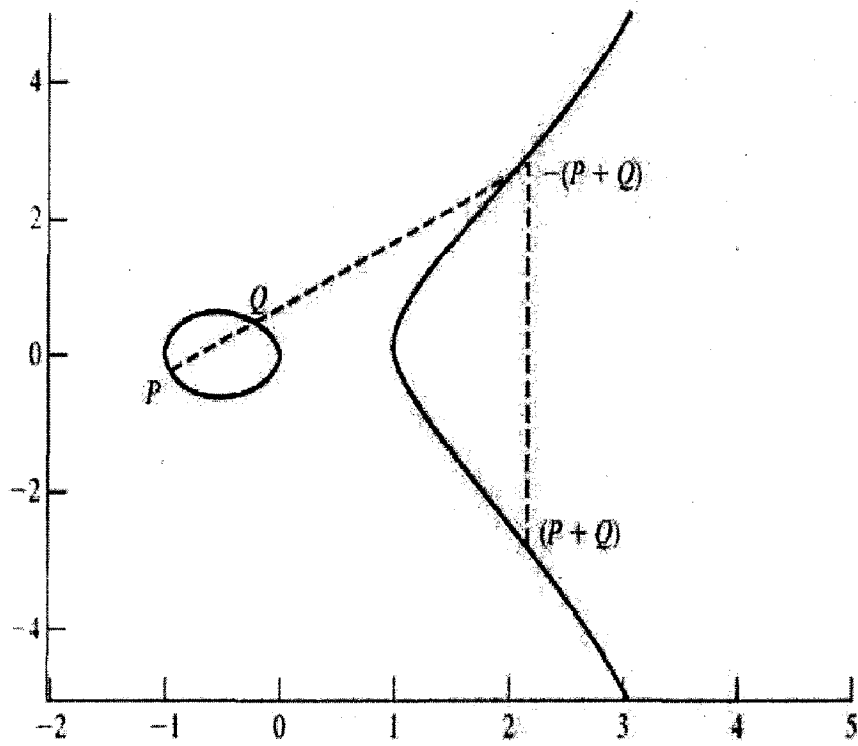


Figure 2.1: Example of elliptic curve: $y^2 = x^3 - x$

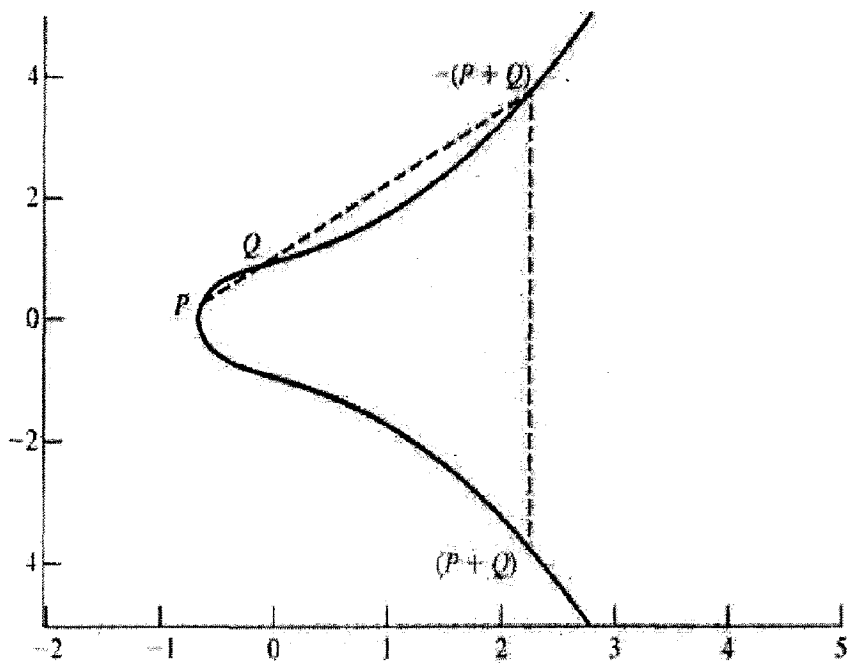


Figure 2.2: Example of elliptic curve: $y^2 = x^3 + x + 1$

2.3 Finite Fields

A finite field consists of a finite set of elements together with two binary operations called addition and multiplication, which satisfy certain arithmetic properties. The order of a finite field is the number of elements in the field. There exists a finite field of order q if and only if q is a prime power. If q is a prime power, then there is essentially only one finite field of order q ; this field is denoted by F_q . There are, however, many ways of representing the elements of F_q . Some representations may lead to more efficient implementations of the field arithmetic in hardware or in software. If $q = p^m$ where p is a prime and m is a positive integer, then p is called the characteristic of F_q and m is called the extension degree of F_q .

2.3.1 Prime Field F_p

Let p be a prime number. The finite field F_p called a prime field, is comprised of the set of integers $\{0, 1, 2, \dots, p-1\}$ with the following arithmetic operations:

- Addition: If $a, b \in F_p$ then $a+b = r$, where r is the remainder when $a+b$ is divided by p and $0 \leq r \leq p-1$ known as addition modulo p .
- Multiplication: If $a, b \in F_p$ then $a \cdot b = s$, where s is the remainder when $a \cdot b$ is divided by p and $0 \leq s \leq p-1$ known as multiplication modulo p .
- Inversion: If a is a non-zero element in F_p , the inverse of a modulo p , denoted by a^{-1} , is the unique integer $c \in F_p$ for which $a \cdot c = 1$.

The equation of the elliptic curve on a prime field F_p is

$$y^2 \bmod p = x^3 + ax + b \quad (2.17)$$

where,

$$4a^3 + 27b^2 \bmod p \neq 0 \quad (2.18)$$

The elements of finite field are integers from 0 to $p-1$. Operations are performed using modular arithmetic. The prime number p is large enough to provide security to cryptosystem. The domain parameters for EC over field F_p are [14]:

- p prime number defined for finite field F_p .
- a, b parameters defining the curve Equation 2.17.
- G generator point (x_g, y_g) a point on the curve to perform operations.
- n order of the elliptic curve. Scalar must be between 0 and $n-1$.
- h cofactor, where $h = \#E(F_p)/n$. $\#E(F_p)$ = number of points on elliptic curve.

2.3.1.1 Algebraic Formulae Over F_p

- $P+O=O+P=P$ for all $P \in E(F_p)$
- If $P=(x, y) \in E(F_p)$ then $(x, y)+(x,-y)=O$. (The point $(x,-y)$ is denoted by $-P$, and is called the negative of P , observe that $-P$ is indeed a point on the curve.
- Point addition: Let $P=(x_1, y_1) \in E(F_p)$ and $Q=(x_2, y_2) \in E(F_p)$, where $P \neq \pm Q$. Then $P+Q=(x_3, y_3)$ where

$$x_3 = \left\{ \frac{y_2 - y_1}{x_2 - x_1} \right\}^2 - x_1 - x_2 \quad (2.19)$$

$$y_3 = \left\{ \frac{y_2 - y_1}{x_2 - x_1} \right\} - x_1 - x_3 - y_1 \quad (2.20)$$

- Point doubling: Let $P=(x_1, y_1) \in E(F_p)$ where $P \neq -P$. Then $2P=(x_3, y_3)$ where

$$x_3 = \left\{ \frac{3x_1^2 + a}{2y_1} \right\} - 2x_1 \quad (2.21)$$

$$y_3 = \left\{ \frac{3x_1^2 + a}{2y_1} \right\}^2 (x_1 - x_3) - y_1 \quad (2.22)$$

Table 2.2 lists the points (other than O) that are part of $E_{23}(1, 1)$. Figure 2.3 plots the points of $E_{23}(1, 1)$; the points, with one exception, are symmetric about $y = 11.5$.

Table 2.2: Points on the Elliptic Curve $E_{23}(1,1)$

(0,1)	(0,22)	(1,7)	(1,16)	(3,10)	(3,13)	(4,0)	(5,4)	(5,19)
(6,4)	(6,19)	(7,11)	(7,12)	(9,7)	(9,16)	(11,3)	(11,20)	(12,4)
(12,19)	(13,7)	(13,16)	(17,3)	(17,20)	(18,3)	(18,20)	(19,5)	(19,18)

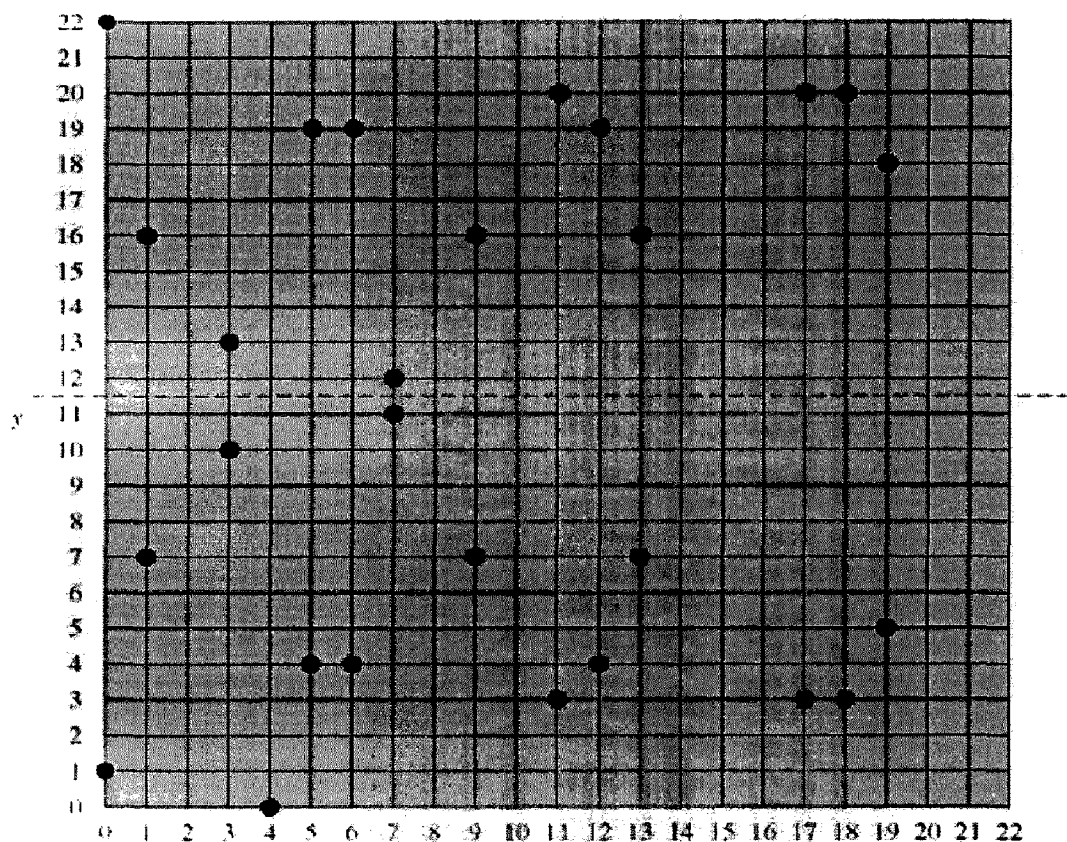


Figure 2.3: The Elliptic Curve $E_{23}(1,1)$

2.3.2 Binary Field F_2^m

The field F_2^m , called a characteristic two finite field or a binary finite field, can be viewed as a vector space of dimension m over the field F_2 which consists of the two elements 0 and 1. That is, there exist m elements $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$ in F_2^m such that each element α can be uniquely written in the form:

$$\alpha = a_0 \alpha_0 + a_1 \alpha_1 + \dots + a_{m-1} \alpha_{m-1}, \quad (2.23)$$

where $a_i \in \{0,1\}$

Such a set $\{\alpha_0, \alpha_1, \dots, \alpha_{m-1}\}$ is called a basis of F_2^m over F_2 . Given such a basis, a field element α can be represented as the bit string $(a_0 a_1 \dots a_{m-1})$. Addition of field elements is performed by bitwise XOR-ing the vector representations. The multiplication rule depends on the basis selected. ANSI X9.62 permits two kinds of bases: polynomial bases and normal bases.

The equation of the elliptic curve on a prime field F_2^m is

$$y^2 + xy = x^3 + ax + b \quad (2.24)$$

where, $b \neq 0$.

The elements of finite field are integers of length atmost m bits. These numbers can be a binary polynomial of degree $m-1$. In the binary polynomial the coefficients can only be 0 or 1. The m is chosen to be large enough to give adequate security. Since the graph is not smooth for this curve so the algebraic rules for point addition and point doubling are adapted. The domain parameters for EC over field F_2^m are [14]:

- m integer defined for finite field F_2^m .
- $f(x)$ irreducible polynomial of degree m used for elliptic curve operations.
- a, b parameters defining the curve Equation 2.24.
- G generator point (x_g, y_g) a point on the curve to perform operations.
- n order of the elliptic curve. Scalar must be between 0 and $n-1$.
- h cofactor, where $h = \#E(F_2^m)/n$. $\#E(F_2^m)$ = number of points on elliptic curve.

2.3.2.1 Algebraic Formulae Over F_2^m

- $P+O=O+P=P$ for all $P \in E(F_2^m)$
- If $P=(x, y) \in E(F_p)$ then $(x, y)+(x, -y) = O$. (The point $(x, -y)$ is denoted by $-P$, and is called the negative of P , observe that $-P$ is indeed a point on the curve.
- (Point addition) Let $P = (x_1, y_1) \in E(F_2^m)$ and $Q=(x_2, y_2) \in E(F_2^m)$, where $P \neq \pm Q$. Then $P+Q=(x_3, y_3)$ where

$$x_3 = \left\{ \frac{y_2+y_1}{x_2+x_1} \right\}^2 + \left\{ \frac{y_2+y_1}{x_2+x_1} \right\} + x_1 + x_2 + a \quad (2.25)$$

$$y_3 = \left\{ \frac{y_2+y_1}{x_2+x_1} \right\} (x_1 + x_2) + x_3 + y_1 \quad (2.26)$$

- (Point doubling) Let $P=(x_1, y_1) \in E(F_2^m)$ where $P \neq -P$. Then $2P=(x_3, y_3)$ where

$$x_3 = x_1^2 \left\{ \frac{b}{x_1^2} \right\} \quad (2.27)$$

$$y_3 = x_1^2 + \left\{ x_1 + \left(\frac{y_1}{x_1} \right) \right\} x_3 + x_3 \quad (2.28)$$

Table 2.3 lists the points (other than O) that are part of $E_{2^4}(g^4, 1)$. Figure 2.4 plots the points of $E_{2^4}(g^4, 1)$.

Table 2.3: Points on the Elliptic Curve $E_{2^4}(g^4, 1)$

$(0,1)$	$(1, g^6)$	$(1, g^{13})$	$(1, g^{13})$	(g^3, g^8)
(g^5, g^3)	(g^5, g^{11})	(g^6, g^8)	(g^3, g^{14})	(g^9, g^{10})
(g^9, g^{13})	(g^{10}, g)	(g^{10}, g^8)	$(g^{12}, 0)$	(g^{12}, g^{12})

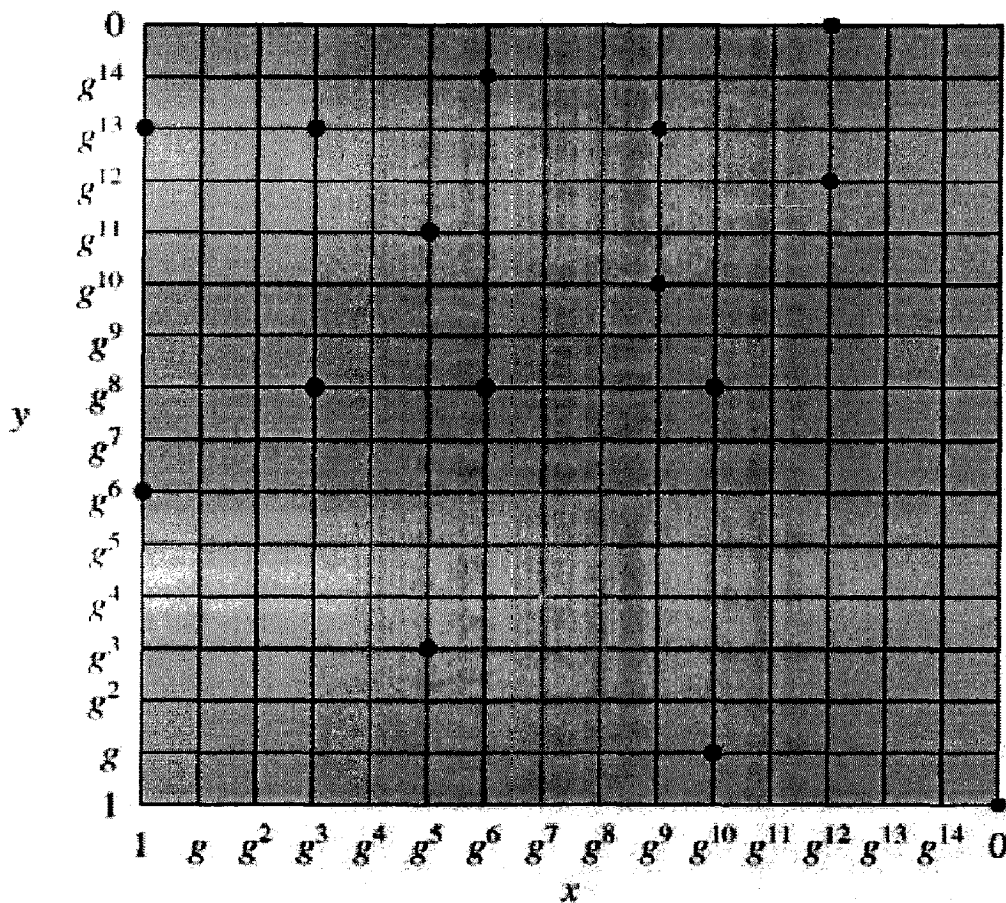


Figure 2.4: The Elliptic Curve $E_{2^4}(g^4, 1)$

2.4 Elliptic Curves Operations over Finite Fields

The main operation is Point multiplication is achieved by two basic elliptic curve operations.

1. Point addition, adding two points J and K to obtain another point L i.e. $L = J + K$, require 1 inversion and 3 multiplication operation.
2. Point doubling, adding a point J to itself to obtain another point L i.e. $L = 2J$, requires 1 inversion and 4 multiplication operation.

2.4.1 Point Addition

Point addition is the addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve.

Consider two points J and K on an elliptic curve as shown in Figure 2.5. If $K \neq -J$ then

a line drawn through the points J and K will intersect the elliptic curve at exactly one more point $-L$. The reflection of the point $-L$ with respect to x-axis gives the point L, which is the result of addition of points J and K. Thus on an elliptic curve $L = J + K$. If $K = -J$ the line through this point intersect at a point at infinity O. Hence $J + (-J) = O$. A negative of a point is the reflection of that point with respect to x-axis [15].

2.4.2 Point Doubling

Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve. To double a point J to get L, i.e. to find $L = 2J$, consider a point J on an elliptic curve as shown in Figure 2.6. If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point $-L$. The reflection of the point $-L$ with respect to x-axis gives the point L, which is the result of doubling the point J, i.e., $L = 2J$. If y coordinate of the point J is zero then the tangent at this point intersects at a point at infinity O. Hence $2J = O$ when $y_j=0$ [15].

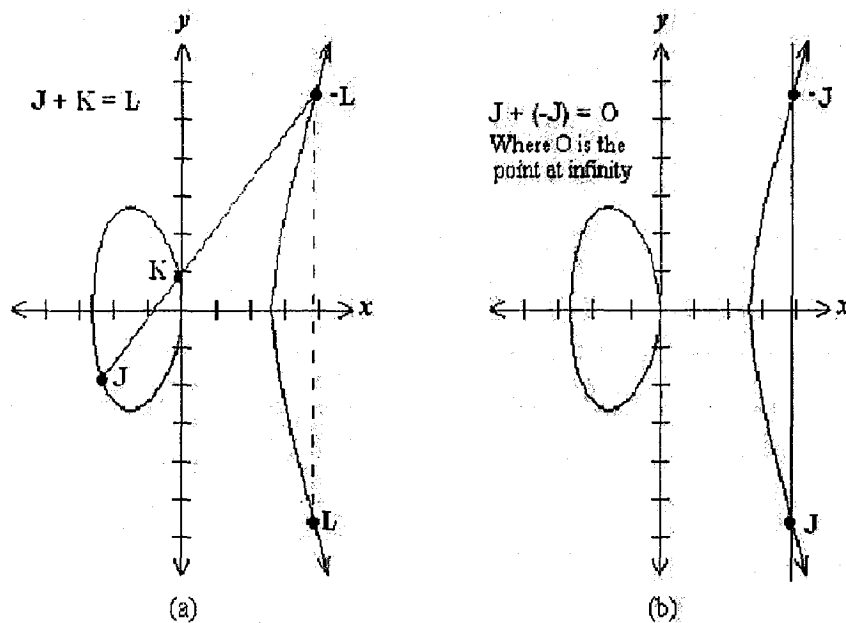


Figure 2.5: Point Addition

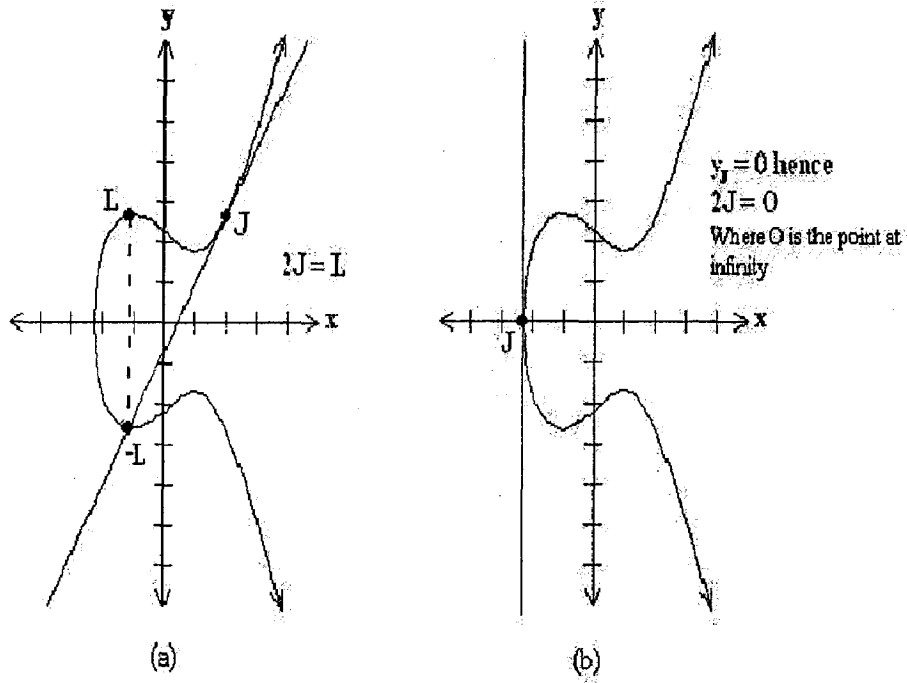


Figure 2.6: Point Doubling

A Smart Card is like an electronic wallet. It is a standard credit card-sized plastic intelligent token within which a microchip has been embedded within its body and which makes it smart. It provides not only memory capacity, but computational capability as well and thus the chip is capable of processing data. This chip holds a variety of information, from stored value used for retail and vending machines to secure information and applications for higher-end operations such as medical/healthcare records. New information and applications can be added depending on the chip capabilities. Smart Cards can store several hundred times more data than a conventional card with a magnetic stripe and can be programmed to reveal only the relevant information. Therefore, unlike the read-only plastic card, the processing power of smart cards gives them the versatility needed to make payments, to configure your cell phones, TVs and video players and to connect to your computers via telephone, satellite or the Internet anytime, anywhere in the world.

3.1 History

The idea of a smart card is much older than most people believe. The first patent for smart cards was filed by two German inventors, Jurgen Dethloff and Helmut Grotrupp, in 1968. The smart card idea was patented in Japan in 1970, in France in 1975 and in the United States in 1978. The first smart card microchip was manufactured in 1977 by Motorola and Bull [16]. The security and other features (size, cost etc.) of the smart card were ideal for embedding smart cards in many applications. The initial high-scale, successful trial of smart cards was performed in France in 1984 by the French Postal and Telecommunication services with the initiation of the first phone chip card [17]. Two years later, millions of smart cards had already been in the market. The chip card was employed in many sectors/industries: telecommunication, financial, identification and others. The most important events for the smart evolution were the introduction of smart cards to the GSM specification and the replacement of the magnetic stripe cards with smart cards by the French financial institutions in the early 1990's.

Today, smart card is the security module protecting investments of millions. A smart card chip acts not only as part of the solution but as an enabler for the whole solution. The industry sectors that exploit smart cards in a great degree are that of telecommunication and banking. The worldwide growth of smart card is shown in Figure 3.1.

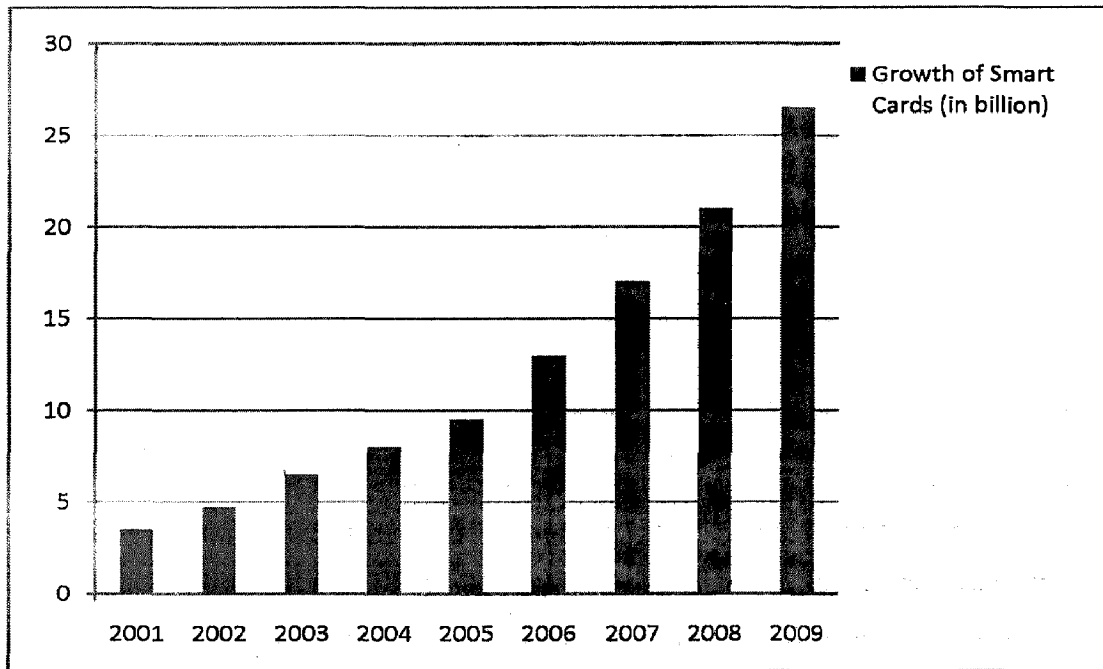


Figure 3.1: Worldwide Growth of Smart Cards

3.2 Types of Smart Cards

Currently, it is common that the term smart card is used to refer to any plastic card carrying a small chip module of specific size that has certain security properties and in addition to memory; it may or not contain a CPU, or microprocessor. In fact, a “true” smart card is a card with an embedded microprocessor and memory. The microprocessor is what makes the smart card “smart” and what allows the card to provide a processing environment. A smart card without an embedded microprocessor is simply a memory card that has a security circuit that prevents unauthorized reading from or writing to the card. A second approach for classifying smart cards is based on the card interface; i.e., the way the smart card communicates with the smart card terminal, or reader. Figure 3.2 gives a transparent illustration about the different types of smart cards.

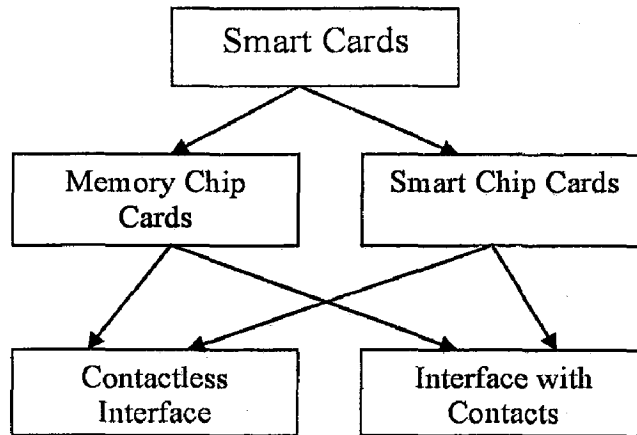


Figure 3.2: Types of Smart Cards

The majority of the smart cards on the market today have between 128 and 1024 bytes of RAM, 1 and 16 kilobytes of EEPROM, and 6 and 16 kilobytes of ROM with the traditional 8-bit CPU typically clocked at a mere 3.57 megahertz. Any addition to memory or processing capacity increases the cost of each card because both are extremely cost sensitive.

3.2.1 Memory Chip Cards

Memory cards can just store data and have no data processing capabilities. These have a memory chip with non-programmable logic, with storage space for data, and with a reasonable level of built-in security. IC memory cards can hold up to 1 – 4 KB of data, but have no processor on the card with which to manipulate that data. They are less expensive than microprocessor cards but with a corresponding decrease in data management security. They depend on the security of the card reader for processing and are ideal when security requirements permit use of cards with low to medium security and for uses where the card performs a fixed operation.

There are two main types of memory chip cards that we have to distinguish: the simple memory chip cards and the “intelligent” memory chip cards. Simple memory chip cards contain non-volatile memory used for storing data and do not provide any particular security features. “Intelligent” memory chip cards contain non-volatile memory for storing data and a security circuit responsible for deciding whether a memory access request is authorized or not.

3.2.2 Microprocessor Cards (Smart Cards)

Microprocessor cards offer greater memory storage and security of data than a traditional magnetic stripe card. Their chips may also be called as microprocessors with internal memory which, in addition to memory, embody a processor controlled by a card operating system, with the ability to process data onboard, as well as carrying small programs capable of local execution. The microprocessor card can add, delete, and otherwise manipulate information on the card, while a memory-chip card (for example, pre-paid phone cards) can only undertake a pre-defined operation. The current generation of smart cards has an eight-bit processor, 32KB read-only memory, and 512 bytes of random-access memory. Data processing permits also the dynamic storage management, which enables realization of flexible multifunctional card. Thus, smart cards have been the main platform for cards that hold a secure digital identity. Hence they are capable of offering advanced security mechanism, local data processing, complex calculation and other interactive processes.

3.2.3 Contact Based Smart Cards

The contact Smart Card has a set of gold-plated electrical contacts embedded in the surface of the plastic on one side. It is operated by inserting the card (in the correct orientation) into a slot in a card reader, which has electrical contacts that connect to the contacts on the card face thus establishing a direct connection to a conductive micro module on the surface of the card. This card, as in Figure 3.3, has a contact plate on the face, which is about ½ inch in diameter on the front, instead of a magnetic stripe on the back like a “credit card”. When the card is inserted into a Smart Card reader, it makes contact with an electrical connector for reads and writes to and from the chip. It is via these physical contact points, that transmission of commands, data, and card status takes place.

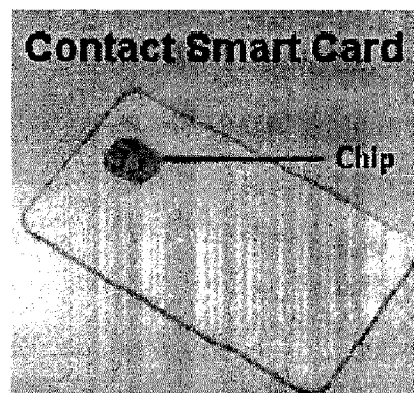


Figure 3.3: Contact Based Smart Cards

3.3 Limitations of Smart Cards

A smart integrated circuit chip is enclosed into a small container called module. This module occupies an area of no more than 25 mm² as specified by the ISO7816 standard. The tiny microchip contains ROM, RAM, EEPROM, a CPU and an operating system. It is obvious that the capabilities of its processing environment are restricted, especially for processing cryptographic elements that usually comprise complicated mathematical transformations.

ROM is fast but has a big disadvantage: it is read only. Applications including cryptographic algorithms cannot be implemented and written in ROM after the card has been manufactured. Hence, smart card application developers cannot use ROM. RAM is very fast and practically useful from the programmer's perspective, but it is uneconomical to have a big size of RAM because it needs large physical space in comparison to the other types of memories (see Figure 3.5: Relative Factor Chip Area). A typical size for RAM in a modern smart card is 16KB. A typical size for EEPROM in a modern smart card is 64KB. This is enough for most of the cryptographic primitives that we are describing in the next section. The problem with EEPROM is that it is much slower than RAM, but usually programmers are by means enforced to use EEPROM where RAM is more appropriate. The reason is the limited size of RAM. Variables containing keys and other information are processed in EEPROM instead in RAM.

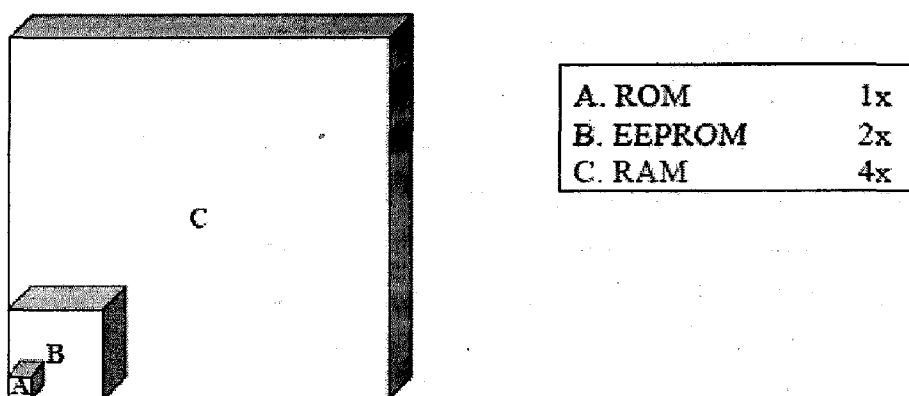


Figure 3.5: Relative Factor Chip Area

CPU has limited processing power. Usually, its clock frequency is no more than 5 MHz and it comprises an 8-bit architecture. The most common architecture is that of personal computers comprising a 32-bit architecture. It is very difficult to achieve an optimized implementation of a cryptographic algorithm on a smart card when the algorithm was designed without considering smart cards' internal architecture. A smart card's operating system provides flexibility to developers, but this flexibility is provided with an extra cost. Cryptographic primitives can be implemented on the top of an operating system using the high-level language the operating system may provide. The high-level language must be compiled to an intermediate language that the operating system understands. The extra overhead emerges from the fact that the operating system has to interpret the intermediate language instructions to hardware instructions. It is a trade-off between performance and the flexibility and error control that an implementation in a high-level language may provide. Some cryptographic primitives may not be able to be implemented on the top of an operating system because of this extra cost. In order to process such numbers in hardware, the microchip may contain a co-processor, or cryptoprocessor especially designed for this purpose. Of course, the co-processor increases significantly the cost of the card and hence is not found in every smart card. Even if a smart card was capable to process data very fast, its capabilities are restricted by the interface transmission rate. Usual transmission rates are less than 40 Kbit/s. Smart cards' tamper-resistance property enables protocol designers to use a smart card as the trusted micro-computer. Smart cards have fewer restrictions year by year, but current restrictions do not prohibit the implementation of cryptographic primitives (on a smart card) that can be used as part of more advanced security protocols.

3.4 Meeting the Implementation Constraints with ECC

Less EEPROM and Shorter Transmission Times

The strength (difficulty) of the ECDLP algorithm means that strong security is achievable with proportionately smaller key and certificate sizes. The smaller key size in turn means that less EEPROM is required to store keys and certificates and that less data needs to be passed between the card and the application so that transmission times are shorter.

Scalability

Smart card applications require stronger security (with longer keys), ECC can continue to provide the security with proportionately fewer additional system resources. This means that with ECC, smart cards are capable of providing higher levels of security without increasing their cost.

No Coprocessor

The nature of the actual computations ³/₄ more specifically, ECC's reduced processing times ³/₄ also contribute significantly to why ECC meets the smart card platform requirements so well. Other public-key systems involve so much computation that a dedicated hardware device known as a *crypto coprocessor* is required. The crypto coprocessors not only take up precious space ³/₄ they add about 20 to 30 percent to the cost of the chip, and about three to five dollars to the cost of each card. With ECC, the algorithm can be implemented in available ROM, so no additional hardware is required to perform strong, fast authentication.

Summarizing, ECC key size advantages afford many benefits for smart cards, and the superior performance offered by Certicom's ECC implementations make applications feasible in low end devices without dedicated crypto hardware [18].

4.1 Overview of Authentication Schemes

Financial transactions and paid services over the web have grown tremendously in the recent times. User authentication is an important security mechanism. Sometimes a key exchange mechanism is also included in the authentication scheme. More often it is not uncommon to verify the identities of the communicating party before allowing him/her to access a remote server or the system's resources. In addition to user authentication, it is also necessary that service provider authenticates themselves to the user to increase consumer confidence. Mutual authentication is required between the communicating parties, prior to any business transaction. Further, the recent developments in the smart card technology and the growing demand for secure applications has lead to lot of research work being done in the area of smart card based systems. Until now, a variety of authentication protocols ranging from complex public-key cryptosystems to simple password based authentication schemes have been proposed.

Mutual authentication between user and the remote server is the essential security aspect that has to be taken into account, and password-based authentication using smart card is one of the simplest and the most widely used strategies. The rapid advance of wireless technology has brought much attention from many researchers who, at the same time, have expressed concerns about security. A remote password authentication scheme is used to authenticate the legitimacy of the remote user over an insecure channel. In such a scheme, the password is often regarded as a secret shared between the authentication server (AS) and the user, and serves to authenticate the identity of the individual login. Through knowledge of the password, the remote user can create a valid login message to the authentication server. AS check the validity of the login message to provide the access rights. Password authentication schemes with smart cards have a long history in the remote user authentication environment.

In 1981, Lamport [19] proposed the first well-known password based remote user authentication scheme without using encryption techniques. However, high hash

overhead and the necessity for password resetting decrease its suitability for practical use. Due to password resetting problem and high computational overheads involved in hash calculations, the suitability of Lamport's scheme for practical use decreases. The Lamport's scheme is not secure, due to some vulnerability. Since then, many similar schemes have been proposed, and each has its pros and cons [20]. With the development of the network services, more and more users need to use the remote resource.

4.2 Authenticated Key Agreement

As we know, the most fundamental security goals are authentication that is a means to verify who is communicating with whom or whether a party is a legitimate one, and confidentiality that is a means to protect messages exchanged over open networks i.e., the Internet. One of the ways to achieve such security goals is to use an authenticated key agreement (AKA) protocol by which the involving parties authenticate each other and then share a common session key to be used for their subsequent secure channels. Up to now, many AKA protocols have been proposed where some take advantage of PKI (Public Key Infrastructure) and others are based on a secret shared between the parties (e.g., human-memorable password).

Recently, various authentication schemes based on elliptic curve cryptosystem (ECC) [21] are proposed to resolve the time-consuming computation problem such as modular exponentiation of traditional public key cryptosystems (PKC) [22] and the limited problem of computation ability and battery capacity of mobile devices.

Generally, the security of ECC is based upon the difficulty of elliptic curve discrete logarithm problem (ECDLP) and elliptic curve Diffie-Hellman problem (ECDHP) [10][11]. Compared with PKC, ECC offers a better performance because it achieves the same security with a smaller key size. For example, 160-bit ECC and 1024-bit RSA have the same security level in practice. However, ECC-based authentication schemes still have some disadvantages:

(1) It needs a key authentication center (KAC) to maintain the certificates for users' public keys like PKC.

- (2) When the number of users is increased, KAC needs a large storage space to store users' public keys and certificates.
- (3) The computation loads and the energy costs of mobile devices very high because users need additional computations to verify the other's certificate [23].

To resolve such problems, several ID-based authentication schemes on ECC are proposed [24]. ID-based authentication schemes on ECC have the following advantages;

- (1) Since the user utilizes his/her unique identity (e.g., name, address, or email address) as his/her public key, the user cannot claim that the authentication information containing his/her identity does not belong to him/her.
- (2) The users do not need to perform additional computations to verify the corresponding certificates without public keys.
- (3) KAC does not need to maintain a large public key table, which is an expensive operation.

4.3 Key Establishment Protocol

Key establishment protocol is a process whereby a shared secret key becomes available to participating entities, for subsequent cryptographic use. It is broadly subdivided into key transport and key agreement protocol [25]. In key transport protocols, a key is created by one entity and securely transmitted to the other entity. In key agreement protocols, both entities contribute information to generate the shared secret key. Key establishment protocol employs symmetric or asymmetric key cryptography.

The goal of any authenticated key establishment protocol is to establish keying data. Ideally, the established key should have precisely the same attributes as a key established face-to-face. However, it is not an easy task to identify the precise security requirements of authenticated key establishment. Several concrete security and performance attributes have been identified as desirable [26].

The fundamental security goals of key establishment protocols are said to be implicit key authentication and explicit key authentication.

4.3.1 Implicit Key Authentication

Let A and B be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. Informally speaking, a key agreement protocol is said to provide implicit key authentication (IKA) (of B to A) if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. A key agreement protocol which provides implicit key authentication to both participating entities is called an authenticated key agreement protocol. A key agreement protocol is said to provide key confirmation (of B to A) if entity A is assured that the second entity B actually has possession of a particular secret key.

4.3.2 Explicit Key Authentication

If both implicit key authentication and key confirmation (of B to A) are provided, the key establishment protocol is said to provide explicit key authentication (EKA) (of B to A). A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation (AKC) protocol. Desirable performance attributes of AKC protocols include a minimal number of passes, low communication overhead, low computation overhead and possibility of pre-computations [26].

4.4 Proposed Scheme based on Authenticated Key Agreement

A password authenticated scheme is proposed and implemented which satisfies the properties such authenticated key agreement, implicit and explicit key confirmation. The scheme is complete and yet simple and can provide adequate security in the todays' scenario of attacks.

The proposed scheme has been divided into 4 phases: **parameter generation phase, registration phase, authentication phase, and password change phase**. First of all, in parameter generation phase, the domain parameters are described and published. These domain parameters are prime number p , elliptic curve equation E , generator point G , and order n . In the registration phase, the server identifies a user and then issues a smart card to the identified user. Then, the user and the server do the authentication phase to authenticate each other and generate a mutually agreed session

key. If the user wants to change his password, he needs to do the password-changing phase. The operations are specific to the application and include scalar multiplication over elliptic curve, hash function based on SHA-1 as specified by FIPS 180-1, and bitwise exclusive-OR operation. The detailed scheme is as follows:

- A. In **parameter generation phase**, AS chooses an elliptic curve E over a finite field F_p such that the discrete logarithm problem is hard in $E(F_p)$. The set of all the points on E is denoted by $E(F_p)$. AS also chooses a point $G \in E(F_p)$ such that the subgroup generated by G has a large order n . AS publishes the parameters (p, E, G, n) .
- B. In **registration phase**, there is a unique identifier, ID associated to each user. AS generates $V = h(ID||K_s) \oplus h(PW)$ and $IM = E_{K_s}(ID||r)$ where PW is initial password selected by the AS, r is a random number to provide identity security and K_s is the private key of AS. AS determines the initial password for U . After receiving the smart card, U is able to immediately change the initial password.
- C. In **authentication phase**, a session key K_{SU} is established. The steps are shown in Figure 4.1. Mutual authentication takes place and the server AS authenticates the card user U and the card user U authenticates the server AS. With the help of three message passing, explicit key confirmation is achieved. Here, both the server AS and the card user U individually calculate the session key based on the parameters passed between them. The session key K_{SU} is established as secret key between the card user and the server.
- D. In **Password-Change Phase** when U wants to change his password PW with a new one, U enters the old password PW , and requests to change password. Then, U enters the new password PW^* . U 's smart card computes $V^* = V \oplus h(PW) \oplus h(PW^*)$, which yields $h(ID||K_s) \oplus h(PW^*)$, and then replaces V with V^* . U can freely change his password and reduces the possibility of the insider attack.

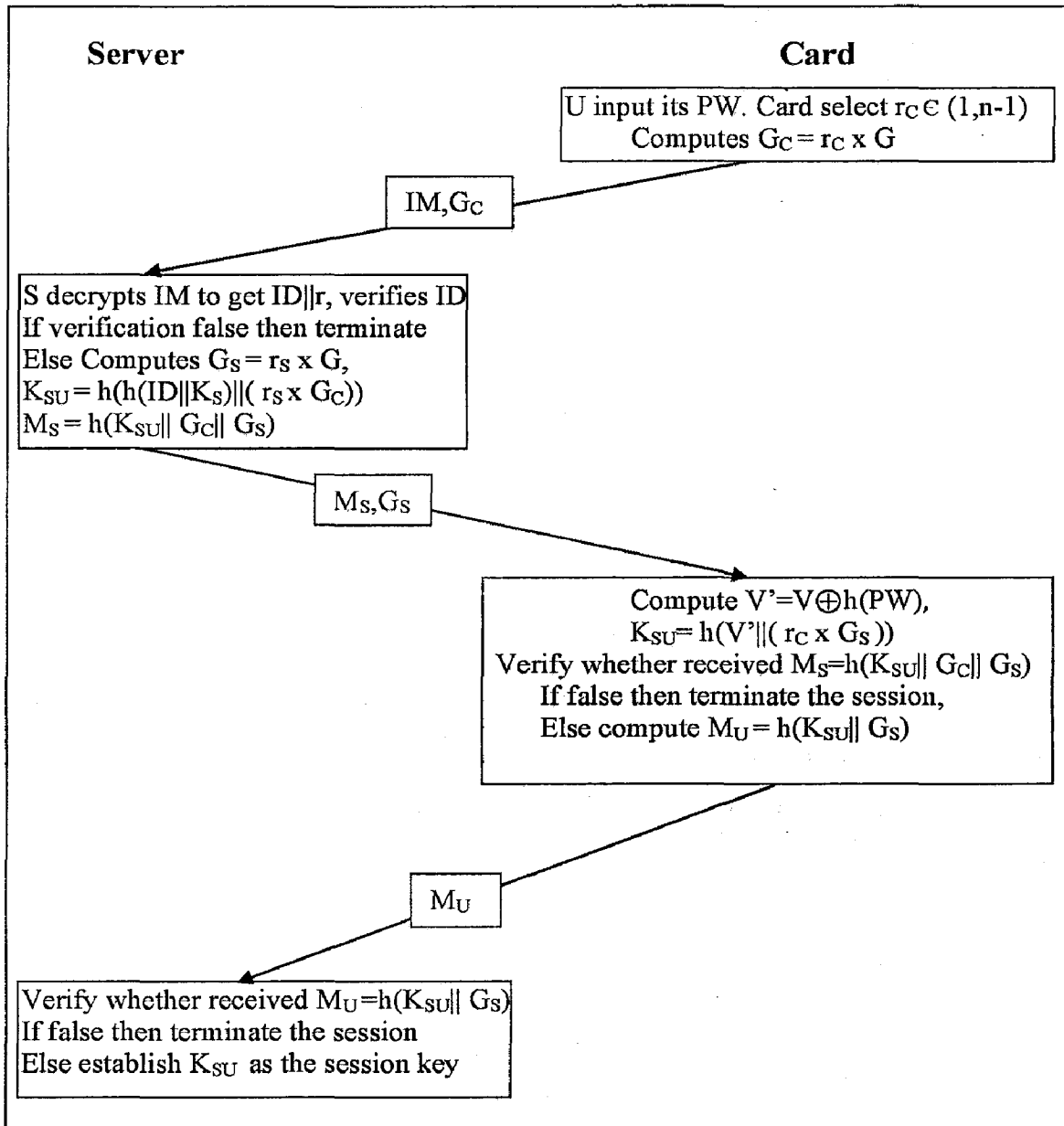


Figure 4.1: Authentication Phase of the Proposed Scheme

Verification

Card computes $K_{SU} = h(h(ID || K_S) || (r_S \times G_C))$, and Server computes $K_{SU} = h(V' || (r_C \times G_S))$ where $V' = V \oplus h(PW)$ and $V = h(ID || K_S) \oplus h(PW)$, thus $V' = h(ID || K_S)$. Now, $r_S \times G_C = r_S \times r_C \times G = r_C \times G_S$. Thus K_{SU} is correct.

Thus, the scheme proposed is quite simple and satisfies the basic and important features of the authenticated key agreement scheme.

The above discussed proposed scheme was implemented on Java Card Kit version 2.2.2 alongwith JDK1.5 running on windows based platform. In this chapter, the implementation detail on Java Card Kit is discussed. The package details as provided by JCRE are discussed briefly alongwith the classes used in the implementation is also discussed briefly.

5.1 Generation of Domain Parameters over Elliptic Curve

As discussed earlier, the elliptic curve discrete logarithm is defined over a finite group. In this scheme, prime field F_p is chosen as recommended by NIST for P-192. The curve specified in [27] uses primes with fast modular reduction algorithms. To select the base point (Generator Point), a point with large prime order n is chosen. The parameters in [27] supply a fixed elliptic curve equation, order n , prime p and a sample base point $G = (G_x, G_y)$.

Hence in this implementation, fixed curve and other domain parameters are chosen as specified in [27]. The chosen domain parameters are given in Appendix[1]. The integers p and r are given in decimal form and others are given in hexadecimal form.

5.2 Java Implementation

The implementation has been done in Java language, on Windows XP platform. JDK 1.5 was used and JCreator was used as an IDE. The reasons for choosing Java as the development language are twofold. Java provides the mechanism to call methods from JCRE through RMI and it has certain built in function providing cryptography constructs. Some of the classes which are used repeatedly are discussed briefly in section 5.4.3.

5.2.1 RMI

The Java Card RMI client side architecture provides the client application mechanisms to initialize and initiate an RMI session with a Java Card applet. It then provides the client application access to the initial remote reference. It also enables

security at transport layer by customizing the message packet during communication with a Java Card technology-compliant smart card to a client application.

5.2.2 Java Card Kit

Java Card Kit 2.2.2 was used to provide support for smart card functionalities. The development kit for the Java Card Platform consists of a suite of tools for designing Java Card technology-based implementations and developing applets. Java Card technology provides a secure environment for applications that run on smart cards and other devices with very limited memory and processing capabilities. Any implementation of a Java Card runtime environment (Java Card RE) contains a virtual machine (VM) for the Java Card platform (Java Card virtual machine), the Java Card Application Programming Interface (API) classes, and support services. The Java Card platform Workstation Development Environment (JCWDE) tool allows the simulated running of a Java Card applet as if it were masked in ROM. It emulates the card environment.

The Java Card WDE is not an implementation of the Java Card virtual machine. It uses the Java virtual machine to emulate the Java Card RE. Class files that represent masked packages must be available on the classpath for the Java Card WDE.

Java programming language source can be converted into APDUs for use on a Java Card technology-enabled smart card. The data flow starts with Java programming language source being compiled and input to the Converter. The Converter tool can convert classes that comprise a Java package to a converted applet (CAP) or to a Java Card technology-based Assembly (Java Card Assembly) file.

A CAP file is a binary representation of converted Java technology package. A Java Card Assembly file can also be used as input to the capgen tool to create a CAP file. CAP files are processed by an off-card installer (scriptgen). This produces an APDU script file as input to the apdutool, which then sends APDUs to a Java Card RE implementation. Figure 5.1 illustrates the components of the installer and how they interact with other parts of Java Card technology.

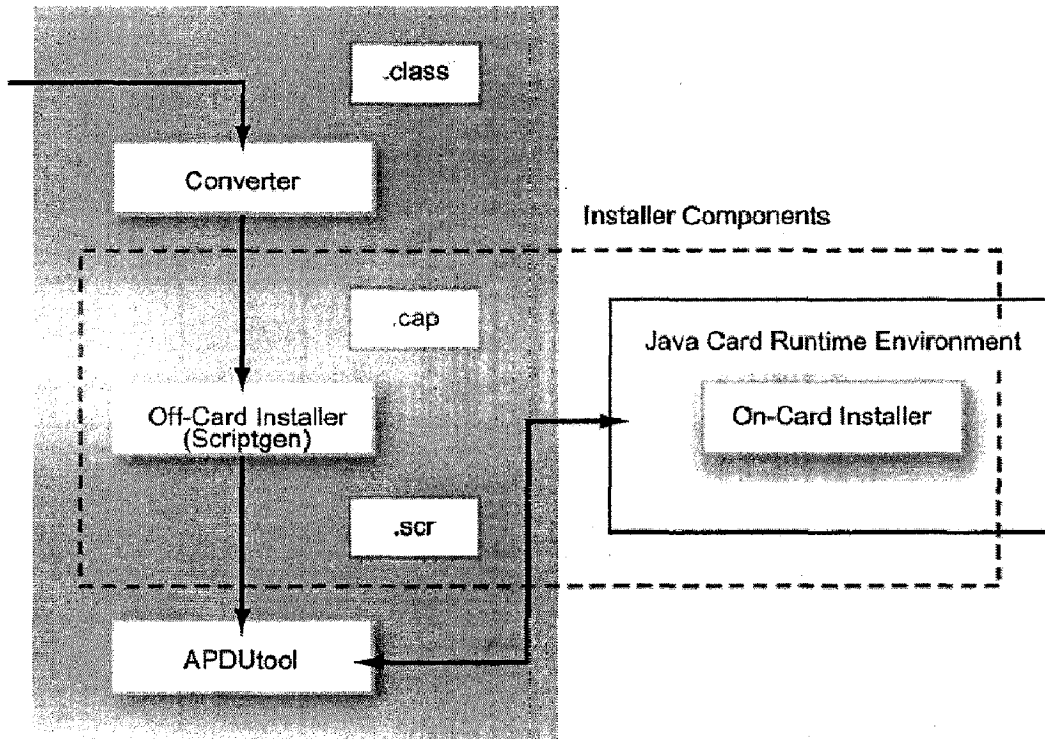


Figure 5.1 Installer Components of Java Card Technology

5.2.3 Package Structure and Classes

The Java Card Kit includes packages for RMI as follows:

- package `com.sun.javacard.clientlib`
 - Interface `CardAccessor` - as an interface, which must be implemented by any custom card accessor.
 - Class `ApduIOCardAccessor` - as a base class or as a source code to implement a custom card accessor.
- package `com.sun.javacard.rmiclientlib`
 - Class `JCRMICConnect` - to select a card applet and to obtain an initial reference.
 - Class `JCCardObjectFactory` Processes the data returned from the card in the format defined for Java CardRMI.

- Class JCCardProxyFactory Processes the data returned from the card in the format defined for Java Card RMI.
- Class JCRemoteRefImpl Represents a reference to a card object
- package java.math
 - Class BigInteger provides modular arithmetic, GCD calculation, primality testing, prime generation, bit manipulation, and a few other miscellaneous operations.

There are certain user defined classes developed particularly for this implementation.

Class ECC: It provides methods for point addition, point doubling and scalar multiplication (i.e. multiplying a point with BigInteger).

Class User: It is developed to carry out the operations as specified in the scheme on behalf of the card.

Class AuthenticationServer: It is developed to carry out the authentication mechanism as specified for the AS in the scheme.

Apart from these classes, there are other classes called wrapper classes to communicate among the defined classes.

After the execution, the CAP file is generated which is in executable binary form. CAP file can be loaded in smart card in form of APDU.

6.1 Security Analysis

The proposed scheme is implemented and various security features can be analyzed. In this section, we will analyze the security of our proposed scheme. The main assumption for guarantee of security lies in:

- 1) The elliptic-curve Diffie–Hellman problem is hard;
- 2) The hash function $h()$ is the pseudorandom permutation for key derivation;

Our scheme can achieve the goal of user authentication and key agreement with great assurance and certainly can prevent the well-known attacks, such as the replay, parallel session, reflection, interleaving, and man-in-the-middle attacks.

Identity Protection: The identifier ID is never explicitly transmitted via the insecure channel. Therefore, both schemes can provide the user's identity protection. Even if the smart card is lost the attacker cannot get the identifier ID in our improved scheme, because he cannot derive the identifier ID from the parameters V and IM without the master secret key K_S .

Replay attack: The replay attack is when an attacker tries to imitate the user to log in to the server by resending the messages transmitted between the user and the server. In our scheme, we use nonces to prevent this kind of attack. In our proposed scheme, the smart card chooses a nonce r_C and computes $G_C = (r_C \times G)$ and then sends it to the server. The second nonce r_S is selected by the server and server computes $G_S = (r_S \times G)$.

Passive attack: A passive attack can be possible if C , the attacker, make a guess at the session key using only information obtainable over network. If the attacker C performs a passive attack, then the session will terminate with both parties accepting. That is, B and A successfully identify themselves to each other, and they both compute the session key. So, C the adversary cannot compute any information about the common shared session key K_S due to the intractability of elliptic curve discrete logarithm problem. Therefore the proposed scheme resists against the passive attack.

Dictionary attack: The dictionary attack could be performed in offline or online mode. An on-line password guessing attack cannot succeed since AS can limit the number of attempts. On the other hand, the offline dictionary attack is very powerful since the attacker does not need to interact with the legitimate entities and can use a lot of computing power. The messages $\{IM, G_C\}$, $\{M_S, G_S\}$, and $\{M_U\}$ of a legitimate authentication session and U's parameter V cannot help the attacker to verify the guessed password, because the corresponding value $r_S \times r_C \times G$ is not available. So the proposed scheme can prevent both types of dictionary attacks.

Smart Card Loss Attack: Suppose user loses his smart card, the adversary cannot use this card without knowing the password of the user. Suppose an adversary wants to change the password, he must know the original password. Thus his attempt to impersonate user fails.

Parallel Session Attack: Suppose an adversary intercepts the message $\{IM, G_C\}$, $\{M_S, G_S\}$, and $\{M_U\}$ to create a valid login. But he cannot succeed as G_C and G_S depends on random r_C and r_S . The adversary cannot find the value of r_C and r_S due to the intractability of elliptic curve discrete logarithm problem.

Explicit Key Confirmation: Using three exchanged messages in the authentication phase, our scheme achieved the explicit key confirmation. AS needs the correct session key K_{SU} to generate the value M_S , which is equal to $h(K_{SU} || G_C || G_S)$. Therefore, AS can be assured that U has actually computed $K_{SU} = h(V^* || (r_C \times G_S))$, after AS has verified that the value M_U is equal to $h(K_{SU} || G_S)$ and thus, U can be assured that AS has actually computed $K_{SU} = h(h(ID || K_S) || (r_S \times G_C))$.

6.2 Comparison with Related Scheme

The proposed scheme is compared on the basis of functionality, storage cost, communication cost and computation cost. In 2008, Juang *et al.* proposed a remote authentication scheme which is also based on ECC but lack some additional security features. In 2005, Fan *et al.* proposed a hash based remote authentication scheme, which is not so secure. The proposed scheme is also compared with the scheme presented by Juang *et al.*[28] and Fan *et al.* [29].

We had made some assumption to do comparison analysis with the scheme of Juang *et al.* Table 6.1 summarizes the security functionalities that are believed to be provided by the Juang *et al.* scheme and our scheme.

Table 6.1: Functionality

Feature \ Schemes	Juang <i>et al.</i> scheme	Fan <i>et al.</i> scheme	Proposed scheme
Password Table	Not required	Not required	Not required
Password	Provided during registration	Provided during registration	Provided during registration
Implicit Key Confirmation	Yes	Yes	Yes
Explicit Key Confirmation	No	No	Yes
Verification Table	Required	Required but of larger size	Required but of smaller size

In the storage cost concern, our scheme requires the smart card to store the parameters V and IM instead of the parameters V , IM , ID , CI , and b in the scheme of Juang *et al.* We can further estimate that the parameters V , IM , ID , CI , and b in the scheme of Juang *et al.* need $128 + 256 + 32 + 32 + 64 = 512$ bits of storage space. In the scheme of Juang *et al.*, AS need about $163 + 128 = 291$ bits of storage space for the secret parameters x and K_S . In case of Fan *et al.* scheme, the parameters are b_i , CID_i , ID_i , n that consume storage of $256 + 96 + 96 + 1024 = 1472$ bits. The server in Fan *et al.* scheme uses a secret key s of 256 bits. Correspondingly, the parameters V and IM in our improved scheme need $128 + 128 = 256$ bits of storage space. AS need a 128-bit storage space for the secret parameter K_S in our scheme. We list the storage costs of the scheme of Juang *et al.* and our scheme in Table 6.2.

Table 6.2: Storage Cost

Entity \ Schemes	Juang <i>et al.</i> scheme	Fan <i>et al.</i> scheme	Proposed scheme
Smart Card	512 bits	1472 bits	256 bits
Server	291 bits (two secret keys x and K_S)	256 bits	128 bits (only one secret key K_S)

Table 6.6: Computation Cost of Proposed scheme

Phases	Entity	Smart Card	Server	Total
Parameter Generation	-	-	-	-
Registration	-	-	$1E + 2H$	$1E + 2H$
Log in	-	$2E_M + 3H$	$2E_M + 3H + 1E$	$4E_M + 6H + 1E$
Password Changing	-	$2H$	-	$2H$
Total	-	$5H+2E_M$	$5H+2E+2E_M$	-----
		Total	$2E + 4E_M + 10H$	

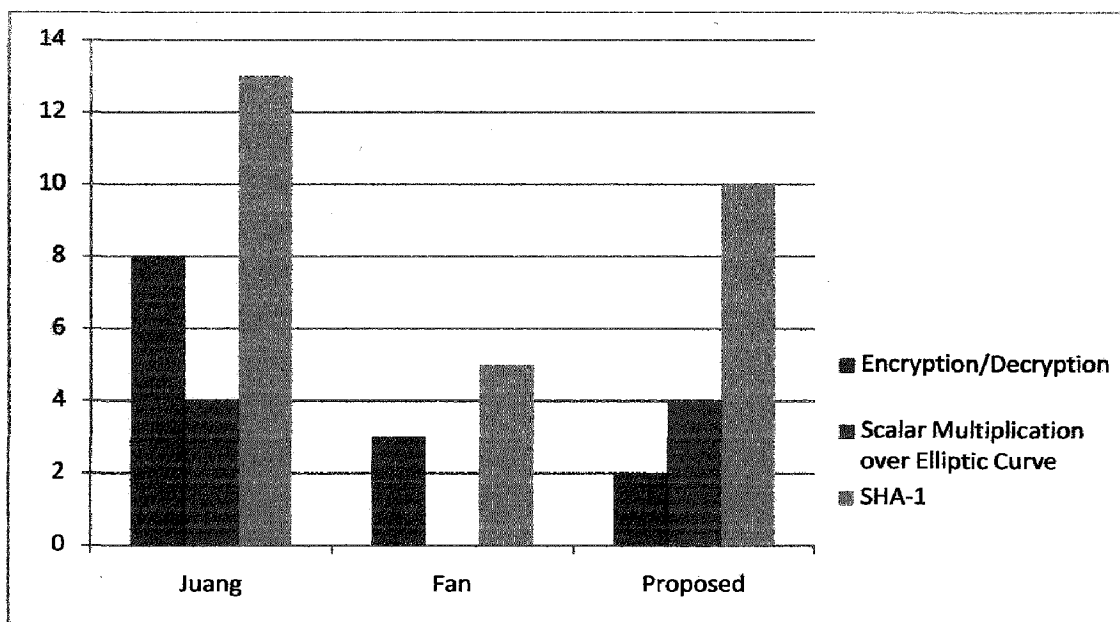


Figure 6.2: Comparative Graph of Computation Complexities

7.1 Conclusion

Public Key Cryptography (PKC) is recently playing an essential role in electronic banking and financial transactions. Elliptic Curve Cryptography (ECC) is one of the best public key techniques for its small key size and high security and is suitable for secure access of smart cards because implementation on smart cards is challenging due to memory, bandwidth, and computation constraints. The tiny smart chip that is embedded on the plastic card is so powerful that can be trusted by a card issuer even when the smart card holders, the end users cannot be trusted. This is the main reason that smart cards has been adopted by the industry and has led to the smart card evolution.

In this dissertation work, we had discussed the ECC providing finite fields for implementation and discussed the point operation over elliptic curves. It is shown that ECC provides increased security due to its underlying mathematical problem. We had also shown the suitability of smart cards for using ECC for remote authentication. We proposed and implemented a password-authenticated key agreement scheme based on ECC. Our scheme provides more guarantees in security as follows: 1) the computation and communication cost is very low; 2) a user can freely choose and change his own password; 3) the privacy of users can be protected; 4) it generates a session key agreed upon by the user and the server; 5) it provides both implicit key and explicit key confirmation; and 6) it can prevent the offline dictionary attack even if the secret information stored in a smart card is compromised. And yet, our scheme is simpler and more efficient for smart card authentication. We had compared the scheme with existing ones and preferable results are obtained and shown.

7.2 Future Work

The proposed scheme of authentication has several scopes of improvements.

1. Point multiplication involves plenty of point addition and point doubling. Each point addition and doubling involves a multiplicative inverse operation. Finding multiplicative inverse is a costly operation in both finite fields, F_p and F_m^2 .

Representing the points in projective coordinate systems can eliminate the need of multiplicative inverse operation in point addition and point doubling and thereby increasing the efficiency of point multiplication operation. For using the projective coordinate in elliptic curve one has to convert the given point in affine coordinate to projective coordinate before point multiplication then convert it back to affine coordinate after point multiplication. The entire process requires only one multiplicative inverse operation. The operation in projective coordinate involves more scalar multiplication than in affine coordinate. ECC on projective coordinate will be efficient only when the implementation of scalar multiplication is much faster than multiplicative inverse operation.

2. Koblitz curves could be used instead of general elliptic curves. This would speed up elliptic curve operations by approximately 50% as shown in [30]. Koblitz curves, also known as anomalous binary curves, are elliptic curves defined over F_m^2 . The primary advantage of these curves is that point multiplication algorithms can be devised that do not use any point doublings. More sophisticated methods of scalar multiplication such as windows- ω method and τ -adic representation for k that has a small number of nonzero digits are in place. These methods can be used to reduce the execution time for performing scalar multiplication.

REFERENCES

- [1] C. L. Hwang, L. J. Chang, Y. S. Yu, "Network-based fuzzy decentralized sliding-mode control for car-like mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 574–585, Feb. 2007.
- [2] G. P. Liu, Y. Xia, J. Chen, D. Rees, W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1282–1297, Jun. 2007.
- [3] C. Lazar, S. Carari, "A remote-control engineering laboratory," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.
- [4] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, L. Benini, "Modeling and optimization of a solar energy harvester system for self-powered wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2759–2766, Jul. 2008.
- [5] F. G. Castineira, F. J. Gonzalez, L. Franck, "Extending vehicular CAN field buses with delay-tolerant networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3307–3314, Sep. 2008.
- [6] P. Marino, F. Poza, M. A. Dominguez, S. Otero, "Electronics in automotive engineering: A top-down approach for implementing industrial field bus technologies in city buses and coaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 589–600, Feb. 2009.
- [7] A. Menezes, P. V. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.
- [8] M. Bellare, P. Rogaway, "Entity authentication and key distribution", *In Proceedings of 13th Advances Cryptology-CRYPTO*, D. R. Stinson, Ed, 1994, LNCS, vol. 773, pp. 232–249.
- [9] M. Bellare, P. Rogaway, "Provably secure session key distribution-The three party case," *In Proceedings of 27th Annual ACM Symp. Theory Computation*, 1995, pp. 57–66.
- [10] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation* 48, 1987, pp. 203-209.

- [11] V. S. Miller, "Use of elliptic curves in cryptography", *In Advances in Cryptology, Proceedings of CRYPTO '85, LNCS, Springer-Verlag*, 1986. Vol. 208, pp. 417-426.
- [12] S. A. Vanstone, "Next generation security for wireless: elliptic curve cryptography", *Computers and Security*, vol. 22, No. 5, Aug. 2003.
- [13] Li F, Xin X, Hu Y, "Identity-based broadcast signcryption", *Computer Standard and Interfaces*, vol. 30, pp. 89-94, 2008.
- [14] Anoop MS, "Elliptic Curve Cryptography-An Implementation Tutorial", Tata Elxsi, India, January 5, 2007.
- [15] J. J. Botes, W.T. Penzhorn, "An implementation of an elliptic curve cryptosystem", *Communications and Signal Processing, COMSIG-94, In Proceedings of the 1994 IEEE South African Symposium*, 1994, pp. 85 -90.
- [16] W. Kim, H. J. Kim, "Smart Cards: Status, Issues, and US Adoption", *Journal of Object Technology*, vol. 3, No. 25, 2004.
- [17] J. Ferrari, "Smart Cards: A Case Study", IBM Corporation, 1998.
- [18] "The Elliptic Curve Cryptosystem for Smart Cards." A Certicom White Paper, May 1998.
- [19] L. Lamport, "Password authentication with insecure communication", *Communication of the ACM*, vol. 24, No. 11, pp. 70-72, 1981.
- [20] D. Jena, S. K. Jena, D. Mohanty, S. K. Panigrahy, "A Novel Remote User Authentication Scheme using Smart Card based on ECDLP", *International Conference on Advanced Computer Control*, 2008.
- [21] Z. G. Chen, X. X. Song, "A distributed electronic authentication scheme based on elliptic curve", *In Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, 2007, pp. 2179-2182.
- [22] R. L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Communications of the ACM*, vol. 21, No. 2, pp. 120-126, 1978.
- [23] E. J. Yoon, K. Y. Yoo, "Robust ID-based Remote Mutual Authentication with Key Agreement Scheme for Mobile Devices on ECC", *In Proceedings of the International Conference on Computational Science and Engineering*, 2009, pp. 633-640.

- [24] X. Cao, W. Kou, L. Dang, B. Zhao, "Identity based multi-user broadcast authentication in wireless sensor networks", *Computer Communications*, vol. 31, pp. 659-667, 2008.
- [25] A. Menezes, P. Van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [26] S. B. Wilson, A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", *In Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), LNCS 1556, Springer-Verlag*, 1999, pp. 339-361.
- [27] NIST, "Recommended Elliptic Curves for Federal Government Use", 1999.
- [28] W. S. Juang, S. T. Chen, H. T. Liaw, "Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards". *IEEE Transactions on Industrial Electronics*, vol. 55, No. 6, 2008.
- [29] C. Fan, Y. Chan, Z. Zhang, "Robust remote authentication scheme with smart cards", *Computer Security*, vol. 24, No. 8, pp. 619-628, 2005.
- [30] K. Jarvinen, J. Skyta, "On parallelization of high-speed processors for elliptic curve cryptography", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2007.

PUBLICATIONS

- [1] Aqeel Khalique, Kuldip Singh, Sandeep Sood, "A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards", *International Journal of Computer Applications*, vol. 2, No. 3, pp. 26-30, 2010.
- [2] Aqeel Khalique, Kuldip Singh, Sandeep Sood, "Implementation of Elliptic Curve Digital Signature Algorithm", *International Journal of Computer Applications*, vol. 2, No. 2, pp. 24-30, 2010.

APPENDIX

1. Domain Parameters by NIST

Following are the ECC domain parameters for 192 bit elliptic curve. These parameters are as given by NIST for use in Federal Government. These parameters are used in the implementation of ECC based scheme in this dissertation work.

Curve P-192

p = 6277101735386680763835789423207666416083908700390324961279
n = 6277101735386680763835789423176059013767194773182842284081
a = -3
b = 64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1
G_x = 188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012
G_y = 07192b95ffc8da78631011ed6b24cdd573f977a11e794811