Proxy Blind Signature using Hyperelliptic Curve Cryptography

Srikanta Pradhan

Roll No. 211CS2286

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Proxy Blind Signature using Hyperelliptic Curve Cryptography

A thesis submitted

in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Computer Science & Engineering

by

Srikanta Pradhan

(Roll 211CS2286)

under the supervision of

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. Sanjay Kuamr Jena
Professor

May 22, 2013

Certificate

This is to certify that the work in the thesis entitled *Proxy Blind Signature using Hyperelliptic Curve Cryptography* by *Srikanta Pradhan*, bearing roll number 211CS2286, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Sanjay Kumar Jena

Acknowledgment

This dissertation, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

The enthusiastic guidance and support of *Prof. Sanjay Kumar Jena* inspired me to stretch beyond my limits. His profound insight has guided my thinking to improve the final product. My solemnest gratefulness to him.

My sincere thanks to *Prof. B.Majhi* and *Prof. S. K. Rath* and for their continuous encouragement and invaluable advice.

It is indeed a privilege to be associated with people like *Prof. A. K. Turuk*, *Prof. B. D. Sahoo,*. They have made available their support in a number of ways.

My humble acknowledgment to *Prof. R. K. Mahapatra*, *Prof. R.K.Dash* and *Prof. M.N. Sahoo* for nourishing my intellectual maturity.

Many thanks to my comrades and fellow research colleagues. It gives me a sense of happiness to be with you all.

Finally, my heartfelt thanks to my family for their unconditional love and support. Words fail me to express my gratitude to my beloved parents, who sacrificed their comfort for my betterment.

Srikanta Pradhan

Abstract

Blind signature is the concept to ensure anonymity of e-coins. Untracebility and unlinkability are two main properties of real coins and should also be mimicked electronically. A user has to fulfill above two properties of blind signature for permission to spend an e-coin.

During the last few years, asymmetric cryptosystems based on curve based cryptographiy have become very popular, especially for embedded applications. Elliptic curves(EC) are a special case of hyperelliptic curves (HEC). HEC operand size is only a fraction of the EC operand size. HEC cryptography needs a group order of size at least $\approx 2^{160}$. In particular, for a curve of genus two field F_q with p $\approx 2^{80}$ is needeed. Therefore, the field arithmetic has to be performed using 80-bit long operands. Which is much better than the RSA using 1024 bit key length. The hyperelliptic curve is best suited for the resource constraint environments. It uses lesser key and provides more secure transmisstion of data.

***Keywords*:** Hyperelliptic curve cryptography, Proxy signature, Blind signature, Symmetric key cryptography, Asymmetric cyptography.

Contents

Chapter 1

Introduction

The study of information hiding and verification is called Cryptography. It includes the protocols, algorithms and strategies to securely and consistently prevents access of sensitive information from unauthorised person and enable verifiability of every component in a communication.

Cryptanalysis is the study of how to circumvent the use of cryptography for unintended recipients or called as code breaking. Cryptography and cryptanalysis are sometimes grouped together under the umbrella coined cryptology, encompassing the entire subject. In practice, cryptography is often used to refer the field as a whole, especially as an applied science. Cryptography is an interdisciplinary subject, drawing from several fields. Before the time of computers, it was very much related to linguistics. Nowadays the emphasis has shifted and cryptography takes extensive use of technical areas of mathematics, those areas collectively known as discrete mathematics. This includes topics from information theory, number theory, statistics, computational complexity and combinatorics. This is also a branch of engineering but an unusual one as it must deal with malevolent opposition, intelligent and active.

1.1 History of cryptography

Until a few decades ago, the information collected by an organization was stored on physical files. The confidentiality of the files was achieved by restricting the access to a few authorized and trusted people in th organization, In the same way, only a few authorized people were allowed to change the contents of files. The availability was achieved by designating at least one person who would have access to the files at all times.

With the advent of computers information storage are now in electronic media. Instead of being stored on physical media, it was stored in computers. The three security requirements, however did not change. The files stored in computers required confidentiality, integrity and availability. The implementation of these requirements however is different and more challenging.

Some security mechanisms can be implemented using cryptography. Cryptography used to refer to the science and art of transforming messages to make them secure and protect from attacks. Although in past cryptography referred only to the encryption and decryption of messages using secret keys, now a days it is defined as involving three distinct mechanisms: symmetric key cryptography and asymmetric key cryptography.

1.2 Symmetric key cryptography

An entity Alice can send a message to another entity Bob over an insecure channel with the assumption that an adversary Eve can not understand the contents of the message by simply eavesdropping over the channel. The original message form Alice to Bob is called plaintext: the message that is sent through the channel is called the ciphertext. To create the ciphertext from the plaintext, Alice uses an encryption algorithm and a shared secret key. To create the paintext from ciphertext. Bob uses a decryption algorithm and the same secret key. A key is a set of values(numbers) that the encryption/decryption algorithms use for operations.

Note that the symmetric key encryption uses a single key(the key itself may be a set of values) for both encryption and decryption. In addition, the encryption and decryption algorithm are inverses of each other. If P is the plaintext, C is the ciphertext, and k is the key, the encryption algorithm $E_k(x)$ creates the ciphertext from the plaintext; the decryption algorithm $D_k(x)$ creates the plaintext form the ciphertext. It is assumed that $E_k(x)$ and $D_k(x)$ are inverses of each other. They cancel the effect of each other if they are applied one after the other on the same input. [1]

$$\begin{aligned} Encryption : C &= E_k(P) \\ Decryption : P &= D_k(C) \\ In\ which,\ D_k(E_k(x)) &= E_k(D_k(x)) = x \end{aligned} \tag{1.1}$$

The popular modern symmetric key cryptography are

1. Data Encryption Standard (DES)
2. Advanceed Encryption Standard (AES)

The following sections describe the mathematics behind the asymmetric key cryptography.

Groups A group **(G)** is a set of elements with a binary operation • that satisfies four properties (or axioms) [1]. A commutative group, also called an **abelian group** if a group in which operator satisfies the four properties for group plus an extra property, commutative. The four properties for group plus commutative are defined as follows:

- **Closure:** If a and b are elements of G, then c=a•b is also element of G.
- **Associativity:** If a,b and c are elements of G, then (a•b)•c = a•(b•c)
- **Commutativity:** For all a and b in G, we have a•b=b•a.
- **Existence of identity:** For all a in G, there exists an element e called the identity element such that e•a=a•e=a

- **Existence of inverse:** For each a in G, there exists an element a', called the inverse of a such that a•a'=a'•a=e

Finite Group A group is called a finite group if the set has a finite number of elements; otherwise, it is an infinite group.

Order of a Group The order of a group —G— is the number of elements in the group.

Subgroups: A subset H of a group G is a subgroup of G if H itself is a group with respect to the operation on G. In other words, if G=$< S, \bullet >$ is a group, H=$< T, \bullet >$ is a group under the same operation and T is a non empty subset of S, then H is a subgroup of H.

Cyclic subgroups: If a subgroup of a group can be generated using the power of an element, the subgroup is called the cyclic subgroup. The term power here means repeatedly applying the group operation to the element:

$$a^n \rightarrow a \bullet a \bullet a \cdots \bullet a \qquad (ntimes)$$

Cyclic Groups: A cyclic group is a group that is its own cyclic subgroup.

Ring: A ring R, denoted $< \{...\}, \bullet, \square >$, is a set of elements with two binary operations. The first operation must satisfy all five properties required for an abelian group. The second operation must satisfy only the first two. In addition, the second operation must be distributed over the first. A commutative ring is a ring in which the commutative property is also satisfied for the second operation.

Field: A field denoted by F=$< \{...\}, \bullet, \square >$, is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse. A finite field is a field with finite number of elements. In Galois field the number of elements should be p^n where p is a prime number and n is a positive integer [1].

1.3 Asymmetric key cryptography

The conceptual differences between symmetric key system and asymmetric key system are based on how these systems keep a secret. The advantages of one can compensate for the disadvantafes of the other. In symmetric key cryptography, the secret must be shared between two persons. In asymmetric key cryptography the secret is personal(unshared); each person creates and keeps his or her own secret.

In a community of n people, $\frac{n(n-1)}{2}$ shared secrets are needed for asymmetric key cryptography; only n personal secret are needed in asymmetric key cryptography.

symmetric key cryptography is based on substitution and permutation of symbols(characters of bits), asymmetric key cryptography is based on applying mathematical function to numbers. In symmetric key cryptography the plaintext and ciphertext are thought of as a combination of symbols. Encryption and decryption permute these symbols or substitute a symbol for another. In asymmetric key cryptography the plaintext and ciphertext are numbers, encryption and decryption are mathematical function that applied to numbers to create other numbers.

In asymmetric key cryptography, the plaintext and ciphertext are numbers; encryption and decryption are mathematical functions that are applied to create other numbers.

Keys: Asymmetric key cryptography uses two separate keys: one private and one public. If encryption and decryption are thought of as locking and unlocking padlocks with keys, then the padlock that is locked with a public key can be unlocked only with the corresponding private key. If Alice locks the padlock with Bob's public key, then only Bob's private key can unlock it.

Plaintext/Ciphertext: Unlike in symmetric key cryptography, plaintext and ciphertext are treated as integers in asymmetric key cryptography. The message must be encoded as an integer(or set of integers) before encryption; the integer(or set of integers) must be decoded into the message after decryption. Asymmetric key cryptography is normally used to encrypt or decrypt small pieces of information

such as the cipher key for ancillary goals instead of message encipherment. However, these ancillary goals play a very important role in cryptography.

Encryption/Decryption: Encryption and decryption in asymmetric key cryptography are mathematical functions applied over the numbers representing the plaintext and ciphertext. The ciphertext can be thought of as C=$F(K_{public}, P)$; the plaintext can be thought of as $P = G(K_{private}, C)$. The function F is used only for encryption; the decryption function G is used only for decryption. Next we show that the function F needs to be a trapdoor one-way function to allow Bob to decrypt the message but to prevent Eve form doing so.

Many algorithm has been proposed as asymmetric key cryptography. But the following are best known for their complex mathmatics, difficult to attack and popular.

1. RSA cryptosystem
2. Rabin cryptosystem
3. ElGamal cryptosystem
4. Elliptic Curve cryptosystem

1.4 Digital signature algorithm

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

To create RSA signature keys, generate an RSA key pair containing a modulus N that is the product of two large primes, along with integers e and d such that

$e \times d \equiv 1(mod\,\varphi(N))$, where φ is the Euler phi-function. The signer's public key consists of N and e, and the signer's secret key contains d.
To sign a message m, the signer computes $\sigma \equiv m^d(modN)$. To verify, the receiver checks that $\sigma^e \equiv m(modN)$.
To prevent attacks, one can first apply a cryptographic hash function to the message m and then apply the RSA algorithm described above to the result. This approach can be proven secure in the so-called random oracle model. Most early signature schemes were of similar type: they involve the use of a trapdoor permutation, such as the RSA function, or in the case of the Rabin signature scheme, computing square modulo composite n. A trapdoor permutation is a family of permutations, specified by a parameter, that is easy to compute in the forward direction, but is difficult to compute in the reverse direction without knowing the private key. However, for every parameter there is a trapdoor (private key) which when known, easily decrypts the message. Trapdoor permutations can be viewed as public-key encryption systems, where the parameter is the public key and the trapdoor is the secret key, and where encrypting corresponds to computing the forward direction of the permutation, while decrypting corresponds to the reverse direction. Trapdoor permutations can also be viewed as digital signature schemes, where computing the reverse direction with the secret key is thought of as signing, and computing the forward direction is done to verify signatures. Because of this correspondence, digital signatures are often described as based on public-key cryptosystems, where signing is equivalent to decryption and verification is equivalent to encryption, but this is not the only way digital signatures are computed.

Usually, this type of signature scheme is vulnerable to a key-only existential forgery attack. To create a forgery, the attacker picks a random signature σ and uses the verification procedure to determine the message M corresponding to that signature. In practice, however, this type of signature is not used directly, but rather, the message to be signed is first hashed to produce a short digest that is then signed. This forgery attack then only produces the hash function output that corresponds

to σ but not a message that leads to that value, which does not lead to an attack. In the random oracle model, this hash-then-sign form of signature is existentially unforgeable, even against a chosen-message attack.

Here RSA cryptosystem is used. We can use different other available cryptosystems.

1.5 Proxy signature

In proxy signature scheme, an original signer delegates its signing capability to a proxy signer, and the proxy signer creates a digital signature on behalf of the original signer. Actually, most of the proposed proxy signature schemes are not feasible in practice because the security of those schemes cannot be really proved without adopting standard signature like DSA. Most of them are not strong, secure, and unbreakable sufficiently against some unknown intentional attacks; in addition, they are not base on standard signature. To conquer those disadvantages, therefore, proxy-protected signature scheme combining standard signature DSA which is pretty well-known by their security properties to reinforce the proxy signature. Combining DSA, proxy signature and PKI mechanism, this work could be used in practice.

The participants are an original signer and a proxy signer. Let p be a prime number and q be a prime division of $p-1$. g is an element of order q in Z_p^*. The tuple (p, q, g) is public and the basic protocol of this scheme uses the following algorithms:

Key generation: The original signer selects a random number $x \in Z_q^*$ as the private key and the corresponding public key is $y = g^x mod p$. Then, the original signer publishes (p, q, g, y).

Proxy key generation: The original signer should do following steps:

1. Select a random number $k_A \in Z_q^*$.

2. Compute $r_A = g^{k_A} \, mod \, p$, and sets $s_A = (x + k_A r_A) \, mod \, q$.

3. Forward (r_A, s_A) to the proxy signer.

On receiving the (r_A, s_A), the proxy signer should verify the validity by checking equation $g^{s_A} = yr_A{}^{r_A} \, mod \, p$. The proxy signer accepts s_A, if the equation holds, and continues following steps. Moreover, the proxy key is s_A.

Proxy signature: The proxy signer can sign a message m on behalf of the original signer to create a signature $S(s_A, m)$ using the proxy key s_A.

Proxy signature verification: The verification of proxy signatures is carried out by using the implicit public key $g^{s_A} = yr_A{}^{r_A} \, mod \, p$ to replace public key in the verification process. The verification is to check if $V(yr_A{}^{r_A}, S(s_A, m), m)$ =True or false.

1.6 Blind signature

Sometimes we have a document that we want to get signed without revealing the contents of the document to the signer. David Chaum has developed some patented blind digital signature schemes for this purpose. The main idea is as follows:

1. Bob creates a blind message and sends to Alice.

2. Alice signs the blinded message and returned the signature on the blinded message.

3. Bob unblinds the signature to obtain a signature on the original message.

Blind signature based on the RSA scheme: Let us briefly describe a blind digital signature scheme developed by David Chaum. Blinding can be done using a variation of the RSA scheme. Bob selects a random number b and calculates the blinded message $B = M \times b^e \, mod n$ where e is Alice's public key and n is the modulus defined in RSA digital signature scheme. Here b can be called a blinding factor. Bob sends B to Alice.

Alice signs the blinded message using the signing algorithm defined in the RSA

digital signature $S_b = B^d \, mod \, n$ where d is Alice's private key. S_b is the signature on the blind version of the message. Bob simply uses the multiplicative inverse of his random number b to remove the blind from the signature. The signature is $S = S_b b^{-1} \, mod \, n$. We can prove that S is the signature on the original message as defined in the RSA digital signature scheme.

$$S \equiv S_b b^{-1} \equiv B^d b^{-1} \equiv (M \times b^e)^d b^{-1} \equiv M^d b^{ed} b{-}1 \equiv M^d b{-}1 \equiv M^d$$

S is the signature if Bob has sent the original message to be signed by Alice.

1.7 Problem statement

Keeping the research directions in view, it has been realised that there exists enough scope to implement hyperelliptic curve in different areas of cryptography. Though hyperelliptic curve cryptography is mainly used to key exchange process, our goal is to implement digital signature algorithm using hyperelliptic curve cryptography.
In particular, the objectives are narrowed to the use of proxy blind signature in hyperelliptic curve cryptography. This proxy blind signature has already been implemented using elliptic curve cryptography. From here we conceived the idea of implementing it on hyperelliptic curve.

1.8 Thesis layout

The rest of the thesis is organized as follows:
Chapter 2: Literature review This chapter contains the abstracts of some selected research papers in the area of hyperelliptic curve cryptography, blind signature and proxy signature. These research papers helped us to gain knowledge regarding the research area.
Chapter 3: Arithmatics of hyperelliptic curve In this chapter all arithmatics are discussed which are needed for hyperelliptic. This chapter decribes about group laws, divisor class group, ideal class group, isomorphisms and isogenies, torison

elements and jacobians.

Chapter 4: Proxy blind signature This chapter includes the proposed scheme. Here we have proposed a proxy blind signature. The different equations are also proved when needed.

Chapter 2

Literature review

2.1 High Performance Arithmetic for special HECC

Regarding the overall speed and power consumption, cryptographic applications in embedded environments like PDAs or mobile communication devices can benefit from specially designed cryptosystems with fixed parameters. Here ahighly efficient algorithm is proposed by them for a hyperelliptic curve cryptosystem (HECC) of genus 2, well suited for their proposed applications on constrained devices. Their work presents a major improvement of HECC arithmetic for certain non-supersingular curves. And these curves are defined over fields of characteristic two. They optimized the group doubling operation and managed to speed up the whole cryptosystem by approximately 27% compared to the previously known most efficient case. Furthermore, an actual implementation of the new formulae on an embedded processor shows its practical relevance. A scalar multiplication can be performed in approximately 50ms on an 800MHz embedded device. [8]

2.2 Parallel Coprocessor Design for Genus-2 HECC

Hardware accelerators are often used in cryptographic applications for speeding up the highly arithmetic intensive public key primitives, e.g. in high-end smart cards. The emerging and very promising public key scheme is based on HyperElliptic Curve Cryptosystems (HECC). Their contribution appears to be the first one to propose architectures for the latest findings in efficient group arithmetic on HECC. The group operation of HECC allows parallelization at different levels: bit-level parallelization and arithmetic operation level parallelization. The authors investigate the trade-offs between both parallelization options and identify the speed and time area optimized configurations. They have found that a coprocessor using a single multiplier ($D = 8$) instead of two or more is best suited. This coprocessor is able to compute group addition and doubling in 479 and 334 clock cycles, respectively. Providing more resources it is possible to achieve 288 and 248 clock cycles, respectively [14].

2.3 Optimal Tower Fields for HECC

Since the development of asymmetric cryptosystems based on elliptic and hyperelliptic curves, it has been a challenging task to implement ECC and HECC over fields of odd characteristic. With the advent of Optimal Extension Fields(OEF), Processor Adequate Finite Fields(PAFF), and more recently OTF, the performance of ECC and HECC over prime(extension) fields increased drastically. For a fixed security level, OTFs offer different field extensions. Thus, the structure of the field can be varied and adapted to the processor word size which yields to an efficient field arithmetic. Our implementation of HECC over OTF shows the practical relevance of OTF in cryptographic implementations. The arithmetic over OTFs allows for a very efficient implementation. On a typical 32-bit embedded microprocessor, namely the ARM7TDMI, a scalar multiplication for a 160-bit group order can be

performed in 44.2ms. Compared to the currently fastest implementation over fields of characteristic two, namely genus-2 HECC over GF(281) on the same processor, this is an improvement of approximately 57%. The implementation on the Pentium 4 processor computes a scalar multiplication over a group order of 160 bit in 2.13ms [15].

2.4 Divisor Class Addition-Subtraction

Here the authors proposed efficient algorithms for the τ-adic sliding window method and applied the algorithms to Koblitz elliptic curve cryptosystem. In this paper, authors extend their ideas to hyperelliptic curve cryptosystem. We give respectively explicit formulae of simultaneous divisor class addition-subtraction algorithm for genus 2 hyperelliptic curves in affine and projective coordinate system and analyze the case of genus 3 hyperelliptic curves. Using this idea and Montgomery trick, we can reduce the number of inversions, multiplications and squares. In addition, they have applied the idea to speed up the precomputation part of two scalar multiplication algorithms for hyperelliptic curve cryptosystem and discuss the efficiency of improved algorithms in detail [5].

2.5 Cantor versus Harley: Optimization and Analysis of Explicit Formulae for HECC

Hyperelliptic curves (HEC) look promising for cryptographic applications. The reason is their short operand size compared to other public-key schemes. The operand sizes seem well suited for small processor architectures, where memory and speed are constrained. However, the group operation has been believed to be too complex and, thus, HEC have not been used in this context so far. In this paper, the authors have tried to increase the efficiency of the genus-2 and genus-3 hyperelliptic curve cryptosystems. For certain genus-3 curves, they have gained

almost 80 percent performance for a group doubling. This work not only improves Gaudry and Harleys algorithm, but also improves the original algorithm introduced by Cantor. Contrary to common belief, they have shown that it is also practical for certain curves to use Cantors algorithm to obtain the highest efficiency for the group operation. In addition, they have introduced a general reduction method for polynomials according to Karatsuba. They have implemented most efficient group operations on Pentium and ARM microprocessors [16].

2.6 Hyperelliptic Curve Crypto Coprocessor

This paper presents a microcode instruction set coprocessor. It is designed to work with an 8-bit 8051 microcontroller which implements a Hyperelliptic Curve Cryptosystem (HECC). The microcode coprocessor is performs a range of Galois Field operations using a dual multiplier/dual adder datapath and storing the intermediate results in the local storage unit of the coprocessor (RAM). This coprocessor is programmed using the software routines from the 8051 microcontroller. It implements the HECC divisors doubling and addition operations. The Jacobian scalar multiplication was computed in a 656 msec (7.87 Mcycles) at 12 MHz clock frequency.

In this paper the authors presented a microcode crypto coprocessor. It is designed to accelerate the Hyperelliptic Curve scalar multiplication using the 8051 microcontroller. The microcode coprocessor is performing the combination of GF(2^{83}) operations. The divisors addition and doubling operations are implemented using software routines based on the coprocessors microcode instructions. The scalar multiplication is developed in C and compiled into 8051 assembly instructions. The total delay of 656 msec (7.8Mcycles) was achieved for the 83-bit HECC scalar multiplication at 12MHz. [9]

2.7 Fast explicit formulae for genus-2 HEC

In accordance with this paper, there was developed a method of arithmetic transforms in Jacobian genus 2 HEC in projective coordinates which provides a lower complexity if compared to the most efficient methods known thus allowing for increase in the efficiency of scalar multiplication. This modification is characterized by:

- reduction of the number of recomputable values
- changes in the sequence of the computational steps
- using dependencies between polynomials in the resultant computation

The suggested modification of method of arithmetic transforms in Jacobian genus 2 HEC results in 3 to 15% reduction of complexity dependant on arithmetic operations used and curve type. Thus, applying the introduced group operation reduces the complexity of the HECC scalar multiplication by 4% compared to the best known formulae [17].

2.8 ID-based Blind Signature

The blind signature scheme is a protocol obtains a signature from a signer, but the signer is unable to read the contents of the message he signs. It is very important technologies in secure e-commerce. The bilinear pairing such as Weil or Tate pairing on elliptic curves and hyperelliptic curves has been found various applications in cryptography. Several ID-based cryptosystems using bilinear pairings were presented. ID-based public key cryptosystem is a good choice for certificate-based public key infrastructures especially when efficient key management and moderate security are required. This paper presents a new ID-based blind signature scheme from bilinear pairings. Also they analyzed their efficiency and security. The scheme proposed by them is more efficient than Zhang and Kim's scheme [18].

2.9 Hybrid Security Model for E-Commerce Channel

The prime requirements for any e-commerce or m-commerce transactions are privacy, authentication,integrity maintenance and non-repudiation. These are the crucial and significant issues in recent times for trade which are transacted over the internet through e-commerce and m-commerce channels. To overcome these issues, various security mechanisms related to symmetric and asymmetric type have been designed. Digital Envelope is one of the practices to attain the prime requirements in e-commerce channels. In this paper, the authors suggest a software implementation of a digital envelope for a secure e-commerce channel that combines the hashing algorithm of MD5, the symmetric key algorithm of AES and the asymmetric key algorithm of Hyperelliptic Curve Cryptography (HECC). The result illustrates that HECC is the best alternative asymmetric key technique rather than ECC and RSA in the digital envelope hybrid cryptosystem [19].

2.10 Ring Signature Scheme Based on Hyper-elliptic Curves

In the paper, authors have analyzed the security problems of ring signature, such as the improper selection of private keys and the low efficiency in software and hardware application. Then they presented an improved ring signature scheme with private key optimization based on HECC. The adaptive optimization and probabilistic encryption of the scheme avoids the indefinite security problem in ring signature and effectively diminish the relevance of different signature generated by the same singer or ring group. So the adversaries can not attack the ring signature or the signature system with effective polynomial algorithms. The trapdoor function of public key cryptosystem in the scheme is based on HCDLP (Hyperelliptic Curve Discrete

Logarithm Problem), and the algorithms make full use of the superiority of HCC, such as short key length, low system overheads etc. The optimization designing strategy reinforces the stability and security of ring signature and also effectively improves the efficiency of group cryptosystem for engineering application [25].

Chapter 3

Arithmatics Of Hyperelliptic Curve

The hyperelliptic curves, which can be seen as a generalization of elliptic curves. In the applications, group elements must be stored and transmitted. For restricted environments or restricted bandwidth it might be useful to use compression even though recovering the original coordinates needs some efforts.
The main emphasis of this chapter is put on the arithmetic properties, i.e., on algorithms to perform the group operation.

For cryptographic purposes on imaginary quadratic hyperelliptic curves given by an equation 3.1.

$$\begin{aligned} C:\ y^2 + h(x)y &= f(x), \\ h, f &\in K[x], \\ deg(f) &= 2g+1, \\ deg(h) &\leq g, f\ \ monic \end{aligned} \tag{3.1}$$

This equation is hyperelliptic curve of genus g over K if no point on the curve over the algebraic closure $\overline{K}$ of K satisfies both partial derivatives $2y + h = 0$ and $f' - h'y = 0$.

The last condition ensures that the curve is nonsingular. The negative of a

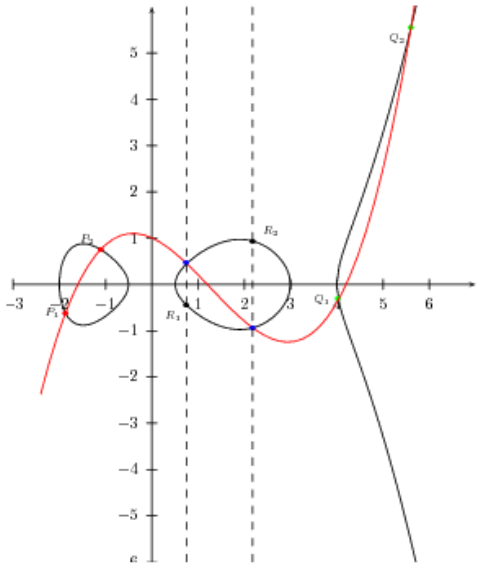

Figure 3.1: Hyperelliptic curve

point $P = (x, y)$ is given by $-P = (x, -y - h(x))$. The points fixed under this hyperelliptic involution are called Weierstraβ points. Elliptic curves are subsumed under this definition as curves of genus one [4].

3.1 Group Laws for hyperelliptic curves

For elliptic curves one can take the set of points together with a point at infinity as a group. For curves of genus larger than one this is no longer possible. The way out is to take finite sums of points as group elements and perform the addition coefficient-wise like $(P+Q) \oplus (R+Q) = P+2Q+R$. This would lead to an infinite group and longer representations of the group elements. The group one actually uses is the quotient group of this group by all sums of points that lie on a function.

Before stating this as a formal definition we give a pictorial description for a genus 2 curve over the reals given by an equation $y^2 = f(x)$ with f monic of degree 5. As for elliptic curves, f is not allowed to have multiple roots over the algebraic closure to satisfy the condition of the definition.

The figure 3.1 demonstrates, this one cannot continue using the chord-and-tangent method from elliptic curves as a line intersects in 5 instead of 3 points. To build a group we take the quotient of the group of sums of points on the curve by the subset of those sums where the points lie on a function, e.g. $R_1 = (x_{R1}, y_{R1})$ and $-R_1 = (x_{R1}, -y_{R1})$ lie on the curve given by $x = x_{R1}$ and hence $R_1 \oplus (R_1) = 0$. Likewise the six points $P_1, P_2, Q_1, Q_2, -R_1, -R_2$ on the cubic add up to zero in the quotient group it concedered.

This way one sees that each element can be represented by at most two points that do not have the same (x, y) coordinate. Namely, any $n > 1$ points give rise to a polynomial of degree $n-1$. There are $\max\{5, 2(n-1)\}$–n other points of intersection. As soon as $n > 2$ the inverse of this sum of points, obtained by inflecting all points at the x-axis, contains fewer points. Repeating this process gives a reduced group element with at most 2 points. The second condition can be seen to hold as points (x_1, y_1) and (x_1, y_1) both lie on the function $x = x_1$. Adding two elements is done in two steps. First the formal sum is formed and then it is reduced. In the general case both group elements consist of 2 points given by $P_1 + P_2$ and $Q_1 + Q_2$ and the 4 points are all different. A function $y = s(x)$ of degree 3 in x passes through all of them having 2 more points of intersection with C. The two new points R_1 and R_2 are inflected and give the result of the addition $(P_1 + P_2) \oplus (Q_1 + Q_2) = R_1 + R_2$.

As in the case of elliptic curves, one can derive the group law from this description by making all steps explicit. If $h \neq 0$ there are still two points with equal $x-$ *coordinate* but the opposite of $P = (x_1, y_1)$ is given by $P = (x_1, -y_1 - h(x1))$ [4].

3.2 Divisor class group and ideal class group

The group we described so far is called the $divisorclassgroup\ Pic_c^0 of C$. To formally define the group law we need to take into account a further point P_∞ called the point at infinity.

Let C be a hyperelliptic curve of genus g over K given by an equation of the

form. The *group of divisors of C of degree 0* is given by equation 3.2

$$\begin{aligned} Div_c^0 = \{D = \sum_{P \in C} n_p P \mid n_p \in \mathbb{Z},\ n_p = 0 \ \ for\ all\ most\ all\ P \in C, \\ \sum_{P \in C} n_p = 0,\ \ and\ such\ that\ \sigma(D) = D\ for\ all\ \sigma \in G_k\} \end{aligned} \tag{3.2}$$

This latter condition means that the divisor is defined over K. This is equivalent to $n_\sigma(P) = n_P$ for all $\sigma \in G_K$, the Galois group of K.

The *divisor class group* $Pic_c^0\, of\, C$ is the quotient group of Div_c^0 by the group of principal divisors, that are divisors of degree zero resulting from functions [4].
Each divisor class can be uniquely represented by a finite sum as given in equation 3.3

$$\sum_{i=1}^{r} P_i - rP_\infty, P_i \epsilon C \left\{P_\infty\right\}, r \leq g \tag{3.3}$$

where for $i \neq j$ we have $P_i = (x_i, y_i) \neq (x_j, -y_j - h(x_j)) = -P_j$.

The following introduces a different representation that is more useful for implementations, and for which one can simply read off the field of definition of the group elements. The theoretical background for this alternative representation is the fact that for hyperelliptic curves the divisor class group is isomorphic to the ideal class group of the function field $K(C)$. Furthermore, the divisor class group is isomorphic to the group of K-rational points of the Jacobian J_C of C. Mumford representation makes explicit this isomorphism and we will use the representation as an ideal class group for the arithmetic. To fix names we keep speaking of the divisor class group and call the group elements divisor classes even when using the notation as ideal classes.

Mumford representation

Let C be a genus g hyperelliptic curve given by $C : y_2 + h(x)y = f(x)$, where $h, f \in K[x]$, $deg\ f = 2g + 1$, $deg\ h \leq g$. Each nontrivial divisor class over K can be

represented via a unique pair of polynomials $u(x)$ and $v(x), u, v \in K[x]$, where

1. u is monic,
2. $deg\, v < deg\, u \leq g$,
3. $u \mid v^2 + vhf$

Let $D = \sum_{i=1}^{r} P_i - rP_\infty$, where $P_i \neq P_\infty$, $P_i \neq -P_j$ for $i \neq j$ and $r \leq g$. Put $P_i = (x_i, y_i)$. Then the divisor class of D is represented by equation 3.4

$$u(x) = \prod_{i=1}^{r}(x - x_i) \tag{3.4}$$

and if P_i occurs n_i times then

$$\left(\frac{\mathrm{d}}{\mathrm{d}x}\right)^j \left[v(x)^2 + v(x)h(x) - f(x)\right]_{|x=x_1} = 0 \;\; for \;\; 0 \leq j \leq n_i - 1 \tag{3.5}$$

A divisor with at most g points in the support satisfying $P_i \neq P_\infty$, $P_i \neq -P_j$ for $i \neq j$ is called a reduced divisor. The first part states that each class can be represented by a reduced divisor. The second part of the theorem means that for all points $P_i = (x_i, y_i)$ occurring in D we have $u(x_i) = 0$ and the third condition guarantees that $v(x_i) = y_i$ with appropriate multiplicity.
We denote the class represented by $u(x)$ and $v(x)$ by $[u(x), v(x)]$. To unify notation we denote the neutral element of the group by [1,0]. There are basically two ways for finding a K-rational divisor class. This can be done by building it from K-rational points on the curve. For instance, choose a random $x_1 \in K$, and try to find $y_1 \in K$ such that

$$y_1^2 + h(x_1)y_1 - f(x_1) = 0$$

In odd characteristic or characteristic zero when $h = 0$, the problem reduces to computing a square root when there exists one. This can be checked with the

Legendre symbol. Already a single point gives rise to a divisor and $[x - x_1, y_1]$ is a valid representative of a divisor class $\bar{D}$. Unless $y_1 = 0$ (or in general ord($\bar{D}$) small) the multiples [n]$\bar{D}$ will have the first polynomial of full degree g for n $\geq$ g. For applications it suggested to implement only the most frequent cases of inputs which implies that the first polynomial u has to have degree g before one can start computing scalar multiples. To build such class of full degree one takes g random distinct points and combines them using Lagrange interpolation to the points $P_1 = (x_1, y_1), \ldots, P_g = (x_g, y_g)$ correspond the polynomials

$$u\,(x) = \prod_{i=1}^{g} (x - x_1) \; and \; v\,(x) = \sum_{i=g}^{t} \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} y_i \tag{3.6}$$

The resulting classes are not completely random as they are built from points defined over K while a K-rational class may also contain points defined over an extension field L/K of degree $[L : K] \leq g$. One chooses a random monic polynomial $u(x) \in K[x]$ of degree g by randomly choosing its g free coefficients. Using the decompression techniques one tries to recover a polynomial v satisfying $u\bar{v}^2 + vhf$. If this fails one starts a new with a different choice of u. The tuple [u, v] represents a divisor class. The amount of work to find v is equal to solving the g quadratic equations in the first approach. However, checking if a u belongs to a class requires more effort. Hence, for an implementation one can trade off the generality of the class for less work.

Using this compact description of the elements, one can transfer the group law that was derived above as a sequence of composition and reduction to an algorithm operating on the representing polynomials and using only polynomial arithmetic over the field of definition K. This algorithm was described by Cantor [CAN 1987] for odd characteristic and by Koblitz [KOB 1989] for arbitrary fields.

Cantor's algorithm

Input: Two divisor classes $\bar{D_1} = [u_1, v_1]$ and $\bar{D_2} = [u_2, v_2]$ on the curve $C : y^2 + h(x)y = f(x)$

Output: The unique reduced divisor D such that $\bar{D} = \bar{D_1} \oplus D_2$ initialization

1 $d_1 \leftarrow gcd(u_1, u_2)$ $[d_1 = e_1u_1 + e_2u_2]$

2 $d \leftarrow gcd(d_1, v_1 + v_2 + h)$ $[d = c_1d_1 + c_2(v_1 + v_2 + h)]$

3 $s_1 \leftarrow c_1e_1,\ s_2 \leftarrow c_1e_2\ and\ s_3 \leftarrow c_2$

4 $u \leftarrow \frac{u_1u_2}{d^2}\ and\ v \leftarrow \frac{s_1u_1v_2+s_2u_2v_1+s(v_1v_2+f)}{d}\ mod\ u$

5 **Repeat**

6 u'$\leftarrow \frac{f-vh-v^2}{u}\ and\ v' \leftarrow (-h-v)\ mod\ u'$ .. u$\leftarrow u'\ and\ v \leftarrow'$

7 **Untill** deg $u \leq g$

8 make u monic

9 **return** [u,v]

3.3 Isomorphisms and isogenies

Some changes of variables do not fundamentally alter the hyperelliptic curve. More precisely, let the hyperelliptic curve C/K of genus g be given by $C : y^2 + h(x)y = f(x)$. The maps
$y \longmapsto u^{2g+1}y' + a_gx'^g + ... + a_1x'^+a_0\ and\ x \longmapsto u^2x' + b\ with\ (a_g, ..., a_1, a_0, b, u) \in k^{g+2} \times K*$
are invertible and map each point of C to a point of $C' : y'^2 + \bar{h}(x')y' = \bar{f}(x')$, where $\bar{h}$, $\bar{f}$ are defined over K and can be expressed in terms of $h, f, a, b, c, d\ and\ u$. Via the inverse map we associate to each point of C' a point of C showing that both curves are isomorphic. These changes of variables are the only ones leaving invariant the shape of the defining equation and, hence, they are the only admissible isomorphisms.

Even if the curves are not isomorphic, the Jacobians of C and C' might share some common properties. One calls J_C and J'_C isogenous if there exists a morphism $\psi\colon J_C \to J'_C$ mapping [1, 0] to the neutral element of J'_C. One important property of

isogenies is that for every isogeny there exists a unique isogeny $\hat{\psi}\colon J'_C \to J_C$ called the dual isogeny such that

$$\hat{\psi} \circ \psi = [n] \ and \ \psi \circ \hat{\psi} = [n]'$$

where $[n]'$ denotes the multiplication by n map on J'_C . The degree of the isogeny ψ is equal to this n. For more background on isogenies.Two curves C/K and C'/K are called isogenous if the corresponding Jacobian varieties J_C and J'_C are isogenous.

3.4 Torsion elements

Definition The kernel of [n] on J_C is denoted by $J_C[n]$. An element $\bar{D} \in J_C[n]$ is called an element of $n - torsion$.
Let C be a hyperelliptic curve defined over K. If the characteristic of K is either zero or prime to n then

$$J_C[n] \simeq (\mathbb{Z}/n\mathbb{Z})^{2g}.$$

Otherwise, when $\mathrm{char}(K) = p$ and $n = p^e$ then

$$J_c[p^e] \simeq (\mathbb{Z}/p^e\mathbb{Z})^r$$

n with $0 \leq, \ for \ all \ e \geq 1$

An elliptic curve E is called supersingular if it has $E[p^e] \simeq P_\infty$, i.e. if it has $p - rank$ 0. A Jacobian of a curve is called supersingular if it is the product of supersingular elliptic curves. Thus especially the p-rank of a supersingular Jacobian variety is 0 but the converse does not have to hold. One also uses the term supersingular curve to denote that the Jacobian of this curve is supersingular.

3.5 Jacobian variety of elliptic curves and group law

Assume that E is an elliptic curve with function field $K(E)$. Hence, Ecan be given as plane projective cubic without singularities and with (at least) one $K - rational$ point P_∞. Clearly $E^1/S_1 = E$.

Let $\bar{D} \in Pic_E^0$ be a divisor class of degree 0, D $\in \bar{D}$ a $K - rational$ divisor. By the RiemannRoch theorem the space $L(D + P_\infty)$ has dimension 1. So there is an effective divisor in the class of $D + P_\infty$ and since this divisor has degree 1 it is a prime divisor corresponding to a point $P \imath E(K)$, and $\phi_K(P) = \bar{D}$. So, $E(K)$ is mapped bijectively to Pic_E^0, the preimage of a divisor class $\bar{D}$ is the point P on E corresponding to the uniquely determined prime divisor in the class of $D + P_\infty$ with $D \in \bar{D}$.
This implies that E is isomorphic to its Jacobian as projective curve. So $E(K)$ itself is an abelian group with the chosen point P_∞ as neutral element, and the addition of two points is given by rational functions in the coordinates in the points. Hence E is an abelian variety of dimension 1 (and vice versa) and we can apply all the structural properties of abelian varieties discussed above to study the structure of $E(K)$ (in dependence of K). This and the description of the addition with respect to carefully chosen equations for E will be among the central parts of the algorithm.

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points with $x_1 \neq x_2$ of the affine curve

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The isomorphism maps them to divisor classes with representatives $D_P = PP_\infty$ and $D_Q = QP_\infty$ of degree 0. The space $L(D_P + D_Q + P_\infty)$ has dimension 1 by RiemannRoch. Hence there exists a function passing through P and Q and having a pole of order at most 1 in P_∞. Such a function is given by the line $l(x, y) = y\lambda x\mu = 0$ connecting P and Q. It has

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \; and \; \mu = y_1 - \lambda x_1$$

As $D_P + D_Q + P_\infty = P + Q \; P_\infty$ has degree 1 and $l \in L(D_P + D_Q + P_\infty)$, there exists an effective divisor in this class that we denote by R and which is a prime divisor. Hence, in the divisor class group we have $\bar{D_P} + \bar{D_Q} = \bar{R} + \bar{P_\infty}$, which is equivalent to $P \oplus Q = R$ on E using the isomorphism from above. Choosing $P \neq Q$ with $x_1 = x_2$ we apply the same geometric construction and get as connecting line the parallel to the $y - axis \; x = x_1$. Hence, the third intersection point has to be interpreted as the point P_∞. This associates to each point $P \in E$ an inverse point $-P$ which has the same $x - coordinate$.
In the remaining case $P = Q$ one can use the considerations above. The function $lL(2PP_\infty)$ passes through P with multiplicity 2, i.e., it is the tangent line to the curve at P . In formulas this means

$$\lambda = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} \; and \; \mu = y_1 - \lambda x_1$$

3.6 Division polynomials

The structure of the group of n-torsion points on E. In that context we showed that for each n there exists a polynomial ψ_n such that the x-coordinates of n-torsion points are the roots of ψ_n. These polynomials are called division polynomials.

If char(K) $\neq 2$, put

$$f_0(x) = 0, \; f_1(x) = 1, \; f_2(x) = 1,$$
$$f_3 = 3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8,$$
$$f_4(x) = 2x^6 + b_2x^5 + 5b_4x^4 + 10b_6x^3 + 10b_8x^2 + (b_2b_8 - b_4b_8)x + (b_4b_8 - b_6^2)$$

Chapter 4

Proxy blind signature scheme using hyperelliptic curve

D. Chaum introduced the concept of a blind signature scheme in 1982. An user A can obtain the signature of B by using this scheme on any given message, without revealing any in formation about the message or its signature. Apart from unforgeability, the scheme ensures untraceability and unlinkability. A lot of work has been done in field of blind signature schemes since Chaum.

For example, in production of coins, the user makes the bank blindly sign a coin using blind signature schemes. The user is in possession of a valid coin such that the bank itself cannot recognize nor link with the user. Whenever a user goes through a valid branch to withdraw a coin, he needs the branch to make proxy blind signature on behalf of the signee bank. This application leads to the need of proxy blind signature schemes [5].

In 1996 Mambo et al [6, 7] introduced the concept of proxy signature. In this scheme an original signer delegates his signing authority to another (proxy) signer in such a way that the proxy signer can sign any message on behalf of the original signer and the verifier can verify and distinguish between normal (original) signature and proxy signature. Here also elaborated the two types of scheme: proxy unprotected

(proxy and original signer both can generate a valid proxy signature) and proxy protected (only proxy can generate a valid proxy signature). These schemes ensures among other things, non-repudiation and unforgeability.

Tan et al. [7] introduced a proxy blind signature scheme, which ensures security properties of both the schemes, viz., the blind signature schemes and the proxy signature schemes. The scheme is based on Schnorr blind signature scheme.

4.1 Proposed scheme: Proxy blind signature

In this section we have presented our proposed scheme. This scheme is based on hyperelliptic curve cryptography and proxy blind signature.
The proposed scheme is depicted as follows
Let a hyperelliptic curve C of genus g be defined over field F_q of finite order defined by equation 4.1

$$y^2 + h(x)y = f(x)\, mod\, q \tag{4.1}$$

Where

$h(x)$is a polynamial and degree of h(x) $\leq$ g

and $f(x)$ is a monic polynamial of degree $\leq$ 2g + 1

The divisor D is defined as follows:

$$D = \sum m_i P_i \tag{4.2}$$

is a formal weighted sum of points P_i of the curve C (and the integers m_i are the wights)
A reduced divisor can be represented as a pair of polynomials $\{u(x), v(x)\}$. Reduced divisors can be added (group addition). e.g. $D_3 = D_1 + D_2$, or doubled (group doubling), e.g. $D_2 = 2D_1 = D_1 + D_1$, and hence the scalar multiplication kD = D +. . . + D for k times is defined. The scalar multiplication kD is the basic operation of HECC.

Parameter initialization

A = Sender Alice

B = Receiver Bob

P = a large prime number

q = large prime factor of (p-1)

g = an element of Z_p^* of order q

x_A = secret key of original signer A

y_A = public key of A = $x_A D$

D = Divisor

4.1.1 Proxy phase:

1. **Proxy generation:** The original signer A randomly chooses k $\in$ Z_q^*, k $\neq$ 1

$$\begin{aligned} R &= kD \\ S &= x_A + k.[D]_x \\ Y_p &= S.D \end{aligned} \tag{4.3}$$

2. **Proxy delivery:** The original signer sends (S,R) to a proxy signer B in a secure way. And makes Y_P public.

3. **Proxy verification:** After receiving the secret key (S,R) the proxy signer B checks the validity of the secret key with the following equation

$$Y_P = y_A + [D]_x \cdot R \tag{4.4}$$

Proof:

$$Y_P = S.D$$

$$= (x_A + k[D]_x)D$$

$$= x_A D + k[D]_x D$$

$$= y_A + [D]_x R$$

If received (S,R) satisfies the equation 4.4 then B accepts it as valid proxy.

4.1.2 Signing phase

1. B chose random number $k_1 \in Z_q^*$ such that $k_1 \neq 1$
   compute : $R_B = k_1 D$
   Now B sends R_B to C

2. C chooses randomly $\alpha, \beta \in Z_q^*$
   $R_c = R_B \parallel \beta Y_p$
   If $R_c = 0$ choose another set of α, β
   else
   e_c=H(r,m)
   e=e_c+β
   C sends e to B

3. B computes S'=$k_1 - Se$
   B sends S' to C

4. C computes
   $S_p = S' + \alpha$

The proxy blind signature is (m,S_p,e_c)

4.1.3 Varificaion:

Recipient of the proxy blind signature computes
e″=h($S_p D \parallel e_c Y_P \parallel M$)

Where Y_P is the public value.

Check e$''$= e_c

If and this statement true then tuple(m,S_P,e_c) is a valid proxy signature.

4.2 Security Analysis

Hyperelliptic curve cryptography is used as the fundamental scheme for this research. The hyperelliptic curve cryptography is more secure than elliptic curve cryptography. It with stand many cryptographic attack. Therefore the security analysis of proxy blind signature is described in following section.

1. A different equation has been used for checking of original signatures and the proxy signatures in our proposed scheme. Thus original signature is distinguishable from the proxy signature.

2. In our scheme to put a valid proxy signature S (in case proxy protected x_B too) is needed. With out knowing X_B or S or both this is impossible to create a valid signature. This is the reason why proxy signature cannot be forged. Furthermore, original signer have no knowledge about x_B though he creates S in case of proxy protected scheme. Hence the proxy signer cannot deny later that the proxy signature not created by him.

3. The public key Y_P has been calculated from the original signers public key y_A. Hence the original signer cannot deny his agreement later. The public key of Proxy signer is also involved in the public key (in case proxy protected). Therefore the proxy signer can be identified from the signature.

Chapter 5

Conclusion

In this thesis we have proposed the proxy blind signature basd on hyperelliptic curve cryptography. Three phases, namely proxy phase, signing phase, varification phase are there in our proposed scheme. In proxy phase the proxy is generated and delivered. In signing phase the signature obtained form previous phase is used to sign. In third phase which is called varification phase, the obtained proxy blind is varified. All these techniques are implemented over hyperelliptic curve cryptography. HECC uses minimum key size less than ECC. This is more suitable than ECC in resource constraint environments.

Bibliography

[1] Behrouz A Forouzan. *Cryptography & Network Security*. McGraw-Hill, Inc., 2007.

[2] J. Pelzl, T. Wollinger, and C. Paar. High performance arithmetic for special hyperelliptic curve cryptosystems of genus two. In *International Conference on Information Technology: Coding Computing, ITCC*, volume 2, pages 513–517, 2004.

[3] A. Hodjat, D. Hwang, L. Batina, and I. Verbauwhede. A hyperelliptic curve crypto coprocessor for an 8051 microcontroller. In *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, volume 2005, pages 93–98, 2005.

[4] Henri Cohen. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. CRC Press, Abingdon, 2005.

[5] Sunder Lal and Amit K Awasthi. Proxy blind signature scheme. 2003.

[6] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto. Proxy signatures for delegating signing operation. In *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 48–57. ACM, 1996.

[7] Seungjoo Kim, Sangjoon Park, and Dongho Won. Proxy signatures, revisited. *Information and Communications Security*, pages 223–232, 1997.

[8] L. You and Y. . Sang. Effective generalized equations of secure hyperelliptic curve digital signature algorithms. *Journal of China Universities of Posts and Telecommunications*, 17(2):100–108+115, 2010.

[9] G. Bertoni, L. Breveglieri, T. Wollinger, and C. Paar. Finding optimum parallel coprocessor design for genus 2 hyperelliptic curve cryptosystems. In *International Conference on Information Technology: Coding Computing, ITCC*, volume 2, pages 538–544, 2004.

[10] S. Baktir, J. Pelzl, T. Wollinger, B. Sunar, and C. Paar. Optimal tower fields for hyperelliptic curve cryptosystems. In *Conference Record - Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 522–526, 2004.

[11] X. Fan and Y. Wang. Simultaneous divisor class addition-subtraction algorithm and its applications to hyperelliptic curve cryptosystem. In *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, volume 1, pages 978–983, 2005.

[12] T. Wollinger, J. Pelzl, and C. Paar. Cantor versus harley: Optimization and analysis of explicit formulae for hyperelliptic curve cryptosystems. *IEEE Transactions on Computers*, 54(7):861–872, 2005.

[13] T. Wollinger and V. Kovtun. Fast explicit formulae for genus 2 hyperelliptic curves using projective coordinates. In *Proceedings - International Conference on Information Technology-New Generations, ITNG 2007*, pages 893–897, 2007.

[14] Z. Zemao, Z. Zhijin, T. Xianghong, and L. Yichun. A new id-based blind signature from bilinear pairings. In *IET Conference Publications*, page 403, 2006.

[15] R. Ganesan and K. Vivekanandan. A novel hybrid security model for e-commerce channel. In *ARTCom 2009 - International Conference on Advances in Recent Technologies in Communication and Computing*, pages 293–296, 2009.

[16] T. J. Park, M. . Lee, K. Park, and K. Chung II. Speeding up scalar multiplication in genus 2 hyperelliptic curves with efficient endomorphisms. *ETRI Journal*, 27(5):617–627, 2005.

[17] P. Gaudry, D. Kohel, and B. Smith. *Counting points on genus 2 curves with real multiplication*, volume 7073 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2011.

[18] D. Schrder and D. Unruh. *Security of blind signatures revisited*, volume 7293 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2012.

[19] I-Te Chen, Ming-Hsin Chang, and Yi-Shiung Yeh. Design of proxy signature in the digital signature algorithm (dsa). *J. Inf. Sci. Eng.*, 22(4):965–973, 2006.

[20] T. Wollinger and C. Paar. Hardware architectures proposed for cryptosystems based on hyperelliptic curves. In *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*, volume 3, pages 1159–1162, 2002.

[21] X. Zhou. Improved ring signature scheme based on hyper-elliptic curves. In *2009 2nd International Conference on Future Information Technology and Management Engineering, FITME 2009*, pages 373–376, 2009.

[22] L. You and F. Zeng. The number of the isomorphism classes of hyperelliptic curves of genus four over finite fields. In *2010 6th International Conference on Information Assurance and Security, IAS 2010*, pages 161–166, 2010.

[23] Nizamuddin, S. Ashraf Ch., W. Nasar, and Q. Javaid. Efficient signcryption schemes based on hyperelliptic curve cryptosystem. In *2011 7th International Conference on Emerging Technologies, ICET 2011*, 2011.

[24] Nizamuddin, Ch Shehzad Ashraf, and N. Amin. Signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In *8th International Conference on High-Capacity Optical Networks and Emerging Technologies, HONET 2011*, pages 244–247, 2011.

[25] T. Satoh. *Generating genus two hyperelliptic curves over large characteristic finite fields*, volume 5479 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2009.

[26] Pierrick Gaudry and Éric Schost. Genus 2 point counting over prime fields. *Journal of Symbolic Computation*, 47(4):368–400, 2012.

[27] A. G. B. Lauder and D. Wan. Computing zeta functions of artin-schreier curves over finite fields ii. *Journal of Complexity*, 20(2-3):331–349, 2004.

[28] WB Lee and CY Chang. Efficient proxy-protected proxy signature scheme based on discrete logarithm. In *Proc. of 10th Conference on Information Security, Hualien, Taiwan*, 2000.