DEVELOPMENT OF CERTIFICATE LESS DIGITAL SIGNATURE SCHEME & ITS APPLICATION IN E-CASH SYSTEM

A Thesis is submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

Pravat Kumar Sahoo (108CS035)

&

Mrutyunjaya Lenka (108CS063)

Department of Computer Science & Engineering
National Institute Of Technology
Rourkela-769008

DEVELOPMENT OF CERTIFICATE LESS DIGITAL SIGNATURE SCHEME AND ITS APPLICATION IN E-CASH SYSTEM

A Thesis is submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

Pravat Kumar Sahoo (108CS035)

&

Mrutyunjaya Lenka (108CS063)

Under The Guidance of

Prof. Sujata Mohanty

Department of Computer Science & Engineering
National Institute Of Technology
Rourkela-769008

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, **“Development of a certificate less digital signature scheme & implementation in e-cash system”** submitted by **Pravat Kumar Sahoo (108CS035)** and **Mrutyunjaya Lenka (108CS063)** in partial fulfilment of the requirements for the award of **Bachelor of Technology Degree** in **Computer Science & Engineering** at National Institute of Technology, Rourkela is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

Prof. Sujata Mohanty
Dept. of Computer Science & Engineering
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENT

We are indebted to our guide **Prof. Sujata Mohanty** for giving us an opportunity to work under her guidance. Like a true mentor, she motivated and inspired us through the entire duration of our work.

Last but not the least, we express our profound gratitude to the Almighty and our parents for their blessings and support without which this task could have never been accomplished.

Pravat Kumar Sahoo
(108CS035)
B. Tech.
Computer Science & Engg.
NIT Rourkela

Mrutyunjaya Lenka
(108CS063)
B. Tech.
Computer Science & Engg.
NIT Rourkela

ABSTRACT

Today's wireless communication systems having limited computational resources and communication bandwidth find certificate less public-key cryptosystems very attractive and vital to their operations in the sense that they help in reducing a significant amount of data load on the network. To eliminate the need of public key digital certificates Shamir proposed ID based cryptosystems in which the user's identity (e.g. name or email id) is used as the public key. However this method had a major drawback of the key escrow problem as a result of which certificate less digital signature (CDS) came into light. The main idea behind CDS is that there's a private key generator (PKG) which generates a partial private key for the user .Then using that key and some of its own private information the user computes its actual private key. PKG's public parameters and the user's private key together calculate the user's public key. Harn, Ren and Lin in 2008 proposed a CDS model which consisted of four generic modules namely PKG, user key generation, signature generation and verification. In this paper, we propose an improvement of the aforesaid CDS scheme in terms of time complexity and signature length and implement the new scheme in an e-cash model proposed by Popescu and Oros. Performance analysis of both the schemes has been carried out in details.

.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION:

Public key cryptosystems are one of the most essential parts of modern communication frameworks. Traditional public key cryptography requires each party which wants to send an encrypted message or a signed document should generate its own public key/private key pair. However the public of an entity needs to be authenticated by means of digital certificate which is provided by a recognized certification authority. However this method of attaching a certificate to authenticate a public key incurs unnecessary bandwidth and computation overhead for a wireless communication device that has a greater limitation in terms of computational power and speed of data transmission. In 1984 Shamir proposed a scheme where a party has to register at a private key generator (PKG) by providing its ID which can be a name or any unique combination of characters. Then the PKG provides a private key to the user or party and the ID of the user can be used as the corresponding public key. The user only needs to know the ID of his partner and the public key of the PKG to send an encrypted/signed document. But this system is not free from key escrow problem which makes the PKG itself as a potential threat of forging the signature or the attacking an encrypted document of one of its users. However a modification to this scheme was made by involving multiple PKGs in generating the user's public key. Then self-certified public key system was proposed by Girault (1991) in which private key is generated by the user while the corresponding public key is calculated using the PKG's and the user's ID. However one still needs digital certificates for fully authenticating the partially authenticated public keys. Al-Riyami and Paterson in 2003 introduced the concept of certificate less digital signature. Their model involved a PKG module which is responsible for generating a partial private key for the user using its own master key. The user then from this private key and some of its own secret parameter calculates the actual private key. Public parameters of PKG and the private key of the user together calculate the user's public key. This method completely removes the key escrow problem as the PKG is not aware of the actual private key of the user. Public keys can be communicated to other users by publishing the same in a public directory or a website. In this way it is also not required to authenticate the public keys by issuing certificates. In 2008 Harn, Ren, Lin [1] proposed a scheme which can convert any DL-based signature scheme into a CDS scheme. According to this scheme any user who wants to produce a signature uses four modules namely; Private Key Generator (PKG), User Key Generation, Signature Generation and Verification.

Information privacy is defined as “an individual’s claim to control the terms under which personal information that is information identifiable to the individual is acquired, disclosed and used”[2]. As technology advances sensitive personal information can be recorded, gathered, analyzed and misused by cyber criminals causing serious damage to customer interests. So it’s an important issue in today’s cyber era to provide information privacy and security to customers who constantly venture into the internet to perform their day to day activities. One of such areas which is most vulnerable to security attack is online transaction systems like e-cash systems used by e-commerce companies. In general three parties are involved in any online transaction; customer, merchant and bank. According to Brickell [3] and Stadler[4] a fair electronic cash system should prevent banks and merchants to obtain vital user information like credit card number, password, transaction history of the customers etc. In cases where there are suspected criminal activities the trusted third party with the help of the bank can revoke the anonymity of the customer or the coin. Popescu and Oros [5] proposed a fair offline e-cash system which implements coin tracing and owner tracing protocol. Trusted third party checks bank’s signature of e-coin and stores the tracing information.

In our paper we propose an improved version of the CDS scheme provided by Harn, Ren, Lin. Our scheme provides signatures of lesser length for input of any file size. Signature generation and verification time are significantly minimized. Then we use this improved CDS scheme in the e-cash model proposed by [5].

This thesis is divided into five chapters. The second chapter gives the fundamental idea behind the scheme. The third chapter briefly describes the existing scheme and our proposed scheme. The performance study gives in details the output of our scheme and the comparison with the previous scheme. Finally in the last chapter we describe how our scheme is implemented in e-cash system. In this paper we have used Popescu and Oros[5] framework for offline e-cash system

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Cryptography

Cryptography means information hiding from unauthenticated persons/programs. It is the application of modern techniques by which modern text (Plain text) is modified in to unintelligible text (cipher text). This technique is otherwise called Encryption. In past cryptography was being done by a common key (Symmetric Key Cryptography) but due to technological advancement now a days we use different key for the encryption process (Asymmetric Key Cryptography). These are described in the next section.

1. Symmetric Key Cryptography
2. Asymmetric Key Cryptography

2.1.1 Symmetric key cryptography

In symmetric key cryptography the sender sends the message by encrypting the message by a key say k_1. The receiver after receiving the cipher text decrypts the message by using the same key k_1. It's assumed here that both the parties use a common key and the transmission of cipher text is done in an insecure channel. This system is flawed if the key k_1 is leaked i.e. if it's known by the adversary.

2.1.2 Asymmetric key cryptography

It's otherwise known as public key cryptosystem or public key encipherment, we have the same situation as of symmetric key cryptosystem, with a few exception. First, there are two keys instead of one, one *public key* and one *private key*. To send a secured message, the sender encrypts with receiver's public key. To decrypt the message the receiver uses his own private key.

- RSA Public Key Cryptosystem

- ElGamal Public Key Cryptosystem

2.2 Digital Signature

A conventional signature is included in the document; it is part of the document. But when we sign a document digitally, we send the signature as a separate document. The sender sends two documents: the message and the signature. The recipient receives both the documents and verifies that the signature belongs to the sender. If it's proven, then accepted else it is rejected.

Several digital signature schemes have evolved during the last few decades. In this section, we discuss some of the widely used signature schemes.

2.2.1 RSA Digital Signature Scheme

In this scheme the signer, first uses an agreed-upon one way hash function to create the digest from the message, $D = h(M)$. Then he/she signs the digest, $S = D^d \bmod n$. The message M and signature S are sent to the receiver. The verifier receives the message and signature. He/she first, uses the sender's public exponent to obtain the digest, $D' = S^e \bmod n$. He/she then applies the Hash algorithm to the message received from the sender to obtain $D = h(M)$ and compares $D \& D'$. If they congruent then message is accepted else rejected.

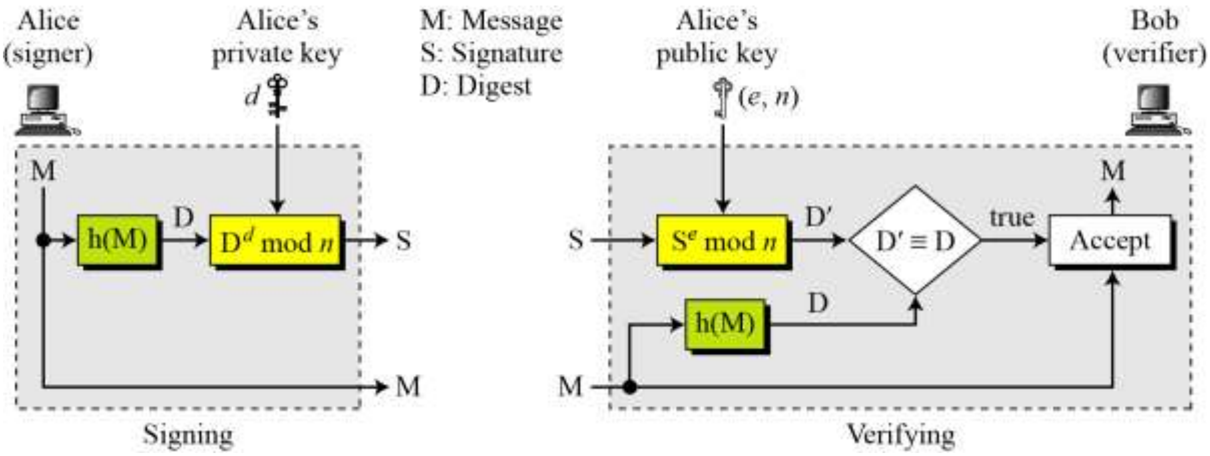


Fig.2.1 RSA Digital signature scheme

2.2.2 Elgamal Digital Signature Scheme

The sender chooses a secret random number r. The sender generates a new random number each time he/she tries to sign a message. Then he/she calculates the signature $S_1 = e_1^r \bmod p$ and $S_2 = (M - d \times S_1) \times r^{-1} \bmod (p-1)$, where S_1 and S_2 are the two signatures. The sender sends M, S_1, S_2 to the receiver. To verify the message, the verifier calculates

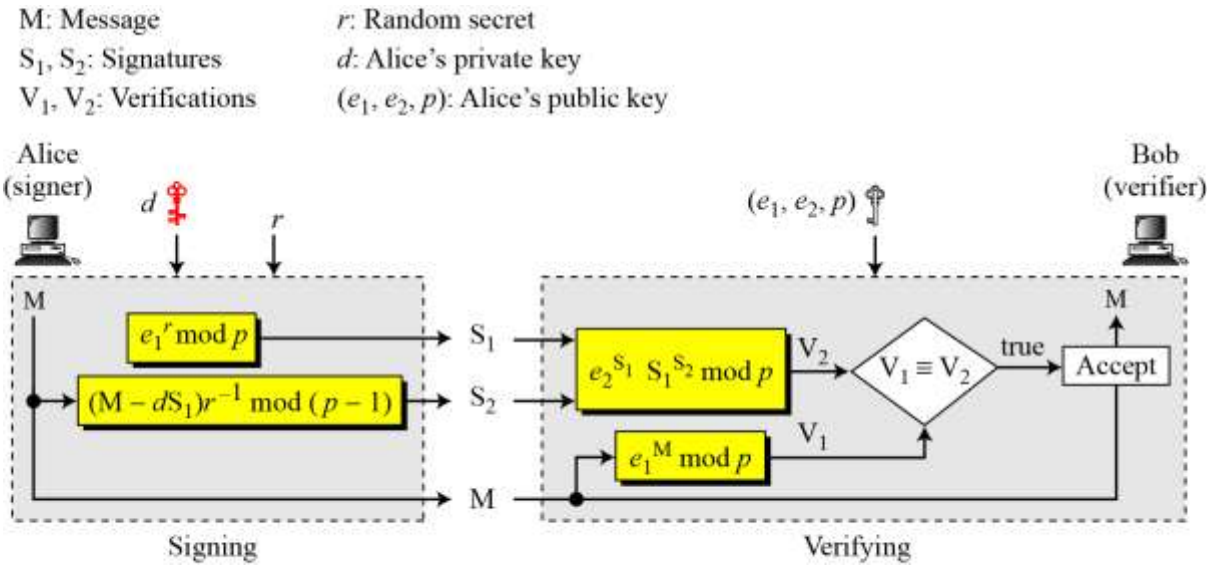


Fig.2.2 ElGamal digital signature scheme

$V_1 = e_1^M \bmod p$ and $V_2 = e_2^{S_1} \times S_1^{S_2} \bmod p$. If $V_1 \equiv V_2$, the message is accepted else rejected.

2.3 Cryptographic Hash Function

A cryptographic hash function creates a fixed-size digest out of a variable-sized message. Creating such a function is best accomplished using iteration. Instead of using a hash function for variable sized inputs, a fixed-sized input is created and is used necessary number of times. This fixed-size input function is compression function.

A set of cryptographic hash functions uses compression function from the scratch. Some of them are described in the following sections:

2.3.1 Message Digest (MD)

Several hash algorithms are designed by Ron Rivest. These are referred as MD2, MD4 and MD5. The MD5 is the strengthened version of MD4 that divides the message into blocks of 512 bits and creates a 128-bit digest. As 128-bit is too small to resist collision attacks we go for what is called SHA (Secure Hash Algorithm).

2.3.2 Secure Hash Algorithm (SHA)

The Secure Hash Algorithm is a standard that was developed by NIST and was published as a FIP standard. It's mostly based on MD5. The standard was revised in 1995, which includes SHA-1. It's then revised to four new versions: SHA-224, SHA-256, SHA-384 and SHA-512. Characteristics of various SHA are shown in Table 1.1.

Characteristics	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Maximum Message size	$2^{64}-1$	$2^{64}-1$	$2^{64}-1$	$2^{128}-1$	$2^{128}-1$
Block Size	512	512	512	1024	1024
Message Digest Size	160	224	256	384	512
Number Of Rounds	80	64	64	80	80
Word Size	32	32	32	64	64

Table 1.1 Characteristics of secure hash algorithms

2.4 Public Key Distribution

In Asymmetric key cryptography, we assume that the public key of any user in the Internet is available to everyone so that they can send messages to each other. But the real problem lies on how to distribute the public keys. In order to distribute public keys we go for digital certificates.

2.4.1 Digital Certificates

A certification authority (CA), a federal or state organization that binds a public key to an entity and issues a certificate. The CA has a well-known public key that cannot be forge. The CA checks the user’s identification and assigns a certificate for his/her public key. Any user wish to send the message can download the certificate and get the public key of the receiver. The ITU has designed X.509, a recommendation that has been accepted by the internet with some changes. X.509 is a way to describe a certificate in a structured way as shown below.

Version Number
Serial Number
Signature Algorithm ID
Issuer Name
Validity period
Subject Name
Subject Public Key
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Signature

Fig.2.3: X.509 Certificate

2.5.Certificate less Signature

Identity-based encryption seems to remove the need for a public key infrastructure, replacing it with the need for a key generation centre that computes a user's private key for them. This is more efficient, but has a significant disadvantage too. The fact that the trusted third party computes the private decryption keys for the users means that that trusted third party can read the messages of every user in the system. There are also significant practical problems associated with identity-based encryption, including the problem of handling key revocation.

In 2003, Al-Riyami and Paterson proposed a new type of encryption scheme that avoids the drawbacks of both traditional public-key encryption and identity-based encryption and in order to resolve the key escrow problem. The new scheme is named as certificate less public key encryption (CL-PKE) because their encryption scheme did not require a public key infrastructure. Certificate less cryptography achieves the best of two worlds. It inherits from identity-based techniques a solution to the certificate management problem in public-key encryption as it is an intermediate between IBC and PKC [Al- Riyami and Paterson 2003]. Its main purpose is to solve the key escrow problem inherited from IBC without the use of certificates as in the traditional PKC. In CL-PKC, a Key Generation Centre (KGC) is involved in issuing user partial private key computed from the master secret key. After that the user also generates an additional user private key and the corresponding user public key of its own. In certificate less cryptography key escrow is seen as an undesirable property, and user encryption and verification keys contain both a user identity and an unauthenticated public key. Similarly, user secret keys are constructed from two partial secrets: one coming from an identity-based trusted authority called the Key Generation Centre (KGC) and another one generated by the user. Certificate less security models capture scenarios where the attacker can be a system user or the KGC itself. To account for the fact that user public keys are not authenticated, attackers are allowed to replace users' public keys to attempt impersonation, a second paper by Al-Riyami and Paterson was published two years later. These authors show that a certificate less signature scheme may be constructed by composing a certificate less KEM with a standard DEM is secure in the Weak Type I and Weak Type II models in a manner similar to Cramer and Shou. Huang and Wong extended the concept of a certificate less KEM to a certificate less Tag-KEM, mirroring the work of Abe et al. in the public key setting. The advantage of Tag-KEMs is that they can be combined with passively secure DEMs and still produce schemes which are fully secure against active attackers. After that the certificate less signature scheme is further improved by Baek et al. In this model, a public key can only be computed after a partial private key has been obtained. This slight change allows Baek, Safavi-Naini and Susilo to propose a certificate less signature scheme based on the CDH problem alone. The only slight drawback of this formulation is that it does not allow messages to be encrypted "into the future". Under the Al-Riyami and Paterson formulation, an entity without knowing the any partial private key can published a public key and therefore they may receive messages that they cannot decrypt until the KGC releases the partial private key to them. Then Liu et al. produced the original model for security against denial of decryption attacks and proposed a generic construction that combined a certificate less encryption .

CHAPTER 3

PROPOSED CERTIFICATE LESS DIGITAL SIGNATURE AND E-CASH SYSTEM

3.1 Review of Existing Scheme

In this scheme the PKG creates a partial private and public key pair, which is sent to the user and the user then calculates its own private and public key pair using DLP based algorithm. Thus the trusted PKG is unaware of the key pair that the user uses. Also it implements the IBS (Identity Based Signature) by using the user's unique ID in producing its private and public key. As we are not using certificates, so in order to distribute the keys the public keys are placed in a public directory or transmitted along the digital signature.

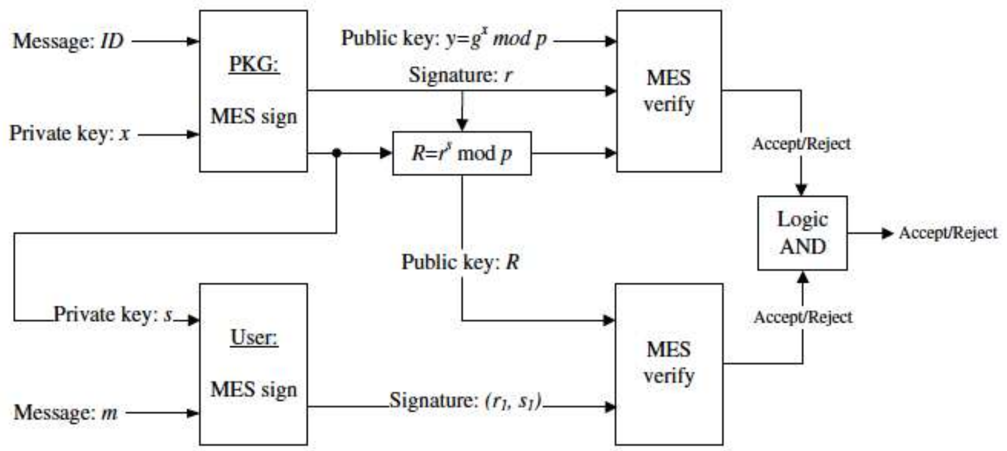


Fig. 3.1: Harn's CDS Scheme

The existing CDS scheme of Harn et al [1] contains four algorithms:

1. PKG key generation
2. User key generation,
3. Message signing
4. Signature verification

3.1.1 PKG Key Generation:

- Generate a large prime p and a generator g of Z_p^*
- Run the random oracle to select a random private key $x \in Z_p^*$

  And computes the public key $y = g^x \bmod p$.
- Return $params = (p, g, y)$ as the public parameters of PKG,

  While keeping x as the master private key.

3.1.2 User key generation

INPUT: $params, ID$

OUTPUT: Public Key Pair (r, R)

- User selects a random private key $v \in Z^*_{p-1}$ with $\gcd(v, p-1) = 1$, and computes $u = g^v \bmod p.\{ID, u\}$ is sent to PKG.
- PKG first selects a $k \in Z^*_{p-1}$ With $\gcd(k, p-1) = 1$ then computes $t = g^k \bmod p$ and $r = u^k \bmod p$.
- PKG solves the linear equation $h(ID, r) = xr + kz \bmod (p-1) \Rightarrow$ $z = k^{-1}(h(ID, r) - xr) \bmod (p-1)$. The output (r, R) is sent to User.
- User computes $s = v^{-1}z \, mod(p-1)$ and $R = r^s \, mod \, p$. s Is each User's private key and (r, R) is each user's public key.

3.1.3 Message signing Sign

INPUT: $params$, message m, private key s

OUTPUT: Signature σ

- Chooses a random $l \in Z^*_{p-1}$ with $\gcd(l, p-1) = 1$ and computes $r_1 = r^l \bmod p$.
- $s_1 = l^{-1}(h(m, r_1) - sr_1) \bmod (p-1)$
- Return $\sigma = (r, R, r_1, s_1)$ as the complete CDS on m.

3.1.4 Signature verification

INPUT: $params$, m, σ

OUTPUT: Accept/Reject

1) $g^{h(ID,r)} = y^r R \bmod p$, and
2) $r^{h(m,r_1)} = R^{r_1} r_1^{s_1} \bmod p$.

If both equations hold, then the CDS will be Accepted, otherwise the CDS will be rejected.

3.2 THE PROPOSED CDS SCHEME

The proposed scheme is having four phases for creating the signature and verifying it. Those are:

1. PKG Key generation
2. User key generation
3. Signature generation
4. Signature Verification.

3.2.1 PKG Key generation:

- p is a large prime and q is a prime factor of $p-1$.
- g is generator belongs to Z_q* of order q.
- PKG chooses its private key $x_A \in Z_q*$
- $y = g^{x_A}$ mod p is public key

3.2.2 User Key generation:

- User selects its private key $x_B \in Z_q^*$ and computes his public key $y_2 = g^{x_B}$ mod

  p
- User sends his ID along with y_2 to PKG.
- PKG selects $k \in Z_q^*$ and computes $r = g^k$mod p
- $s = (k + r \times x_A)$mod p and sends s to user along with ID.

3.2.3 Signature Generation:

- After receiving s, user chooses $l \in Z_q^*$and computes $u = g^l$ mod p
- Then he computes $t = H(m, u^{x_B \times l^{-1}} \times {y_B}^{s \times x_B^{-1}}$ mod $p)$
- Certificate less Signature on message m is $\sigma = (t, s)$.

3.2.4 Signature Verification:

- ○ Any verifier obtaining signature σ on message m can verify the authenticity by checking $t' = H(m, y_B \times g^s$ mod $p)$

If t' and t are equal then the CDS is accepted otherwise it is rejected

3.3. E-Cash System:

Electronic cash (e-cash) is a popular system since it realizes the digitalization of traditional cash system. This scheme enables customers to pay electronic money to merchants for different goods. There are some features that this system must implement, such as:

- *Anonymity:* The merchant/the spender must remain anonymous.

- *Unreusability:* The digital cash/e-cash cannot be reused or copied i.e. to reduce the risk involved in forgery and to establish authentication.
- *Un-forgebility:* Only the authentic users can produce the e-coin.
- *Off-line payment:* Transaction can be offline, i.e. no communication with the bank involved.

Electronic payment is one of the key issues of ecommerce development and many schemes has been propsed till date, but as far as the use of certificateless signature concerned to roboust the Electronic payment security aspect; a lot potential still to be exploited. .First Chaum suggested the electronic cash system in 1982.Subsequently, numerous untraceable electronic cash protocols were proposed based on these constructs (Chaum 1983, Fan and Lei 1998, Ferguson 1994, Pointcheval and Stern 1997, Camenisch et al. 1995, Pointchval and Stern 1996)[7] .

The framework proposed incorporates all these features and is described as follows. It has four parties or entities which are *Customer, The Bank, Trusted Third Party (TTP) & Merchant.* The communications among these entities is shown in fig. 5.1. All these parties maintain some parameters i.e. their private and public key. Here the TTP works as if it's the PKG. It produces the partial private and public keys and using these keys the user produces its own key pair. Thus we avoid the key escrow problem that persisted in traditional schemes. Basically the e-cash system

System parameters :

The system parameters consist of a large prime p, a large prime factor q of $p-1$ and an element $g \in \mathbf{Z}^*_{\mathbf{p}}$ of order q.

The Trusted Third Party

The trusted third party executes the following to set up his parameters.

- Select random secret $\mathbf{x_t} \in \mathbf{Z_q}$
- Calculate $\mathbf{y_t} = \boldsymbol{g}^{\mathrm{x_t}} (\boldsymbol{mod\ p})$
- The public key of the trusted third party is $\mathbf{y_t}$
- The corresponding secret key is $\mathbf{x_t}$

The Bank

The bank executes the following to set up his parameters.

- Select random secret $\mathbf{x_b} \in \mathbf{Z_q}$
- Calculate $\mathbf{y_b} = \boldsymbol{g}^{\mathrm{x_b}} (\boldsymbol{mod\ p})$
- The public key of the bank is $\mathbf{y_b}$
- The corresponding secret key is $\mathbf{x}_{\boldsymbol{b}}$

The Customer

The customer executes the next steps to set up his parameters.

- Select random secret $\mathbf{x_u} \in \mathbf{Z_q}$
- Calculate $\mathbf{y_u} = \boldsymbol{g}^{\mathrm{x_u}} (\boldsymbol{mod\ p})$
- The public key of the customer is $\mathbf{y_u}$

The Payment Protocol

The payment protocol involves the customer and the merchant in which the customer pays the electronic coin to the merchant.

Withdrawal Protocol

The customer contacts the Bank, requesting for a coin. The bank proves the customer's identity and produces a coin represented by the tuple ($\mathbf{c}, \mathbf{r_b}, \mathbf{s_b}, \mathbf{r_t}, \mathbf{s_t}$).

Deposit Protocol

Involves the *Merchant* and the *Bank* as follows (the merchant deposits his electronic coins to the bank):

- The merchant sends the e-cash ($\mathbf{c}, \mathbf{r_b}, \mathbf{s_b}, \mathbf{r_t}, \mathbf{s_t}$) to the bank.
- The bank verifies the validity of the e-coin using the same operations as the merchant.

- The bank checks whether the coin has been double spent. If the coin was not deposited before the bank accepts the coin and will deposit the e-cash to the account of the customer. Then the merchant sends the goods to the customer.

The Customer Tracing Protocol

It involves the *Bank* and the TTP. This protocol is used to determine the identity of the customer in a specific payment transaction. Money laundering can be prevented from detecting illegal customer in this protocol.

- The bank sends the e-coin $(\mathbf{c}, \mathbf{r_b}, \mathbf{s_b}, \mathbf{r_t}, \mathbf{s_t})$ to the trusted third party.
- The trusted third party verifies the validity of the e-coin using the same operations as the merchant and then sends $\bar{\mathbf{r}}$ to the bank. Note that $\bar{\mathbf{r}}$ is linked with the coin **c**.
- The bank can find the corresponding customer from his database (saved in the withdrawal protocol).

The Coin Tracing Protocol

The coin tracing protocol involves the bank and the trusted third party. This protocol determines the e-coin in the case when blackmailing occurs. The blackmailing can be prevented in this protocol

- The customer sends his identity, ID, to the bank.
- The bank sends $\bar{\mathbf{r}}$ to the trusted third party.
- The trusted third party finds the corresponding coin **c** and then sends the coin **c** to the bank.
- The bank can reject the coin **c**.

CHAPTER 4

PERFORMANCE EVALUATION

4.1 PERFORMANCE STUDY

It is clear from the following tables that our scheme has a lesser signature generation and verification time. The signature length is nearly same for different file sizes as we are using a hash function i.e. SHA 1 for the hashing. We have tested the Algorithm for various files and succeeded in generating and verifying the signature. Also the proposed scheme has less computational complexity as compared to the existing scheme.

4.2 Comparison with Existing Scheme

Operation	Signature Generation	Signature Verification
Exponential	$3T_E$	T_E
Hash	T_H	T_H
Multiplication	$3T_M$	$3T_M$

Tab.4.1 Number of operations in proposed scheme

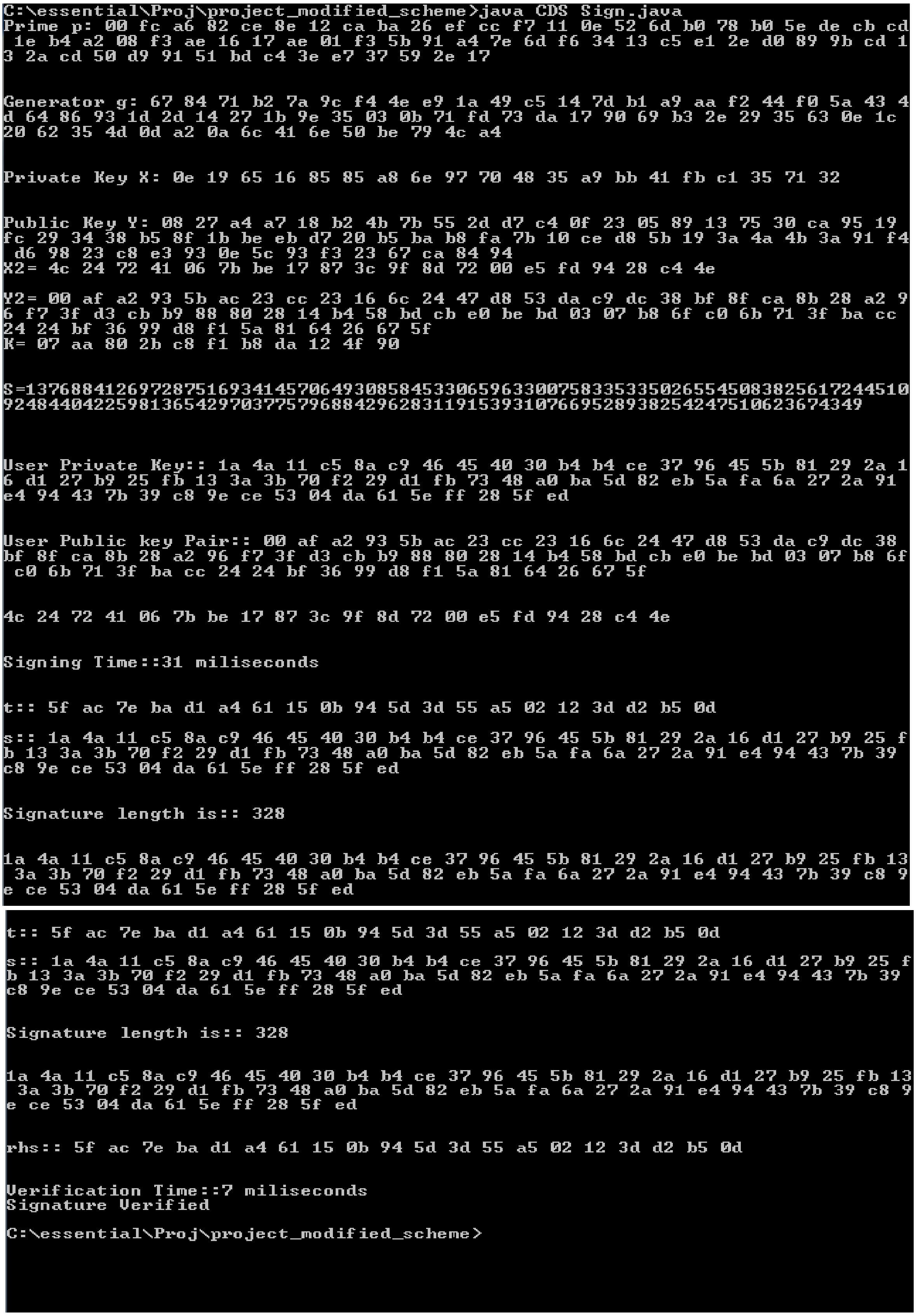
Operation	Proposed Scheme	Harn's Scheme
Exponential	$3T_E$	$7T_E$
Hash	$2T_H$	$3T_H$
Multiplication	$4T_M$	$4T_M$

Tab.4.2 Comparison with existing CDS

T_E - Time taken for Exponential Operation
T_M - Time taken for Multiplication operation
T_H - Time taken for Hash operation

Operation	Existing Scheme (Execution time)	Proposed Scheme (Execution time)
Signature Generation	44 milliseconds	21 milliseconds
Signature Verification	20 milliseconds	5 milliseconds

Tab.4.3 Comparison of existing scheme and proposed scheme



```
C:\essential\Proj\project_modified_scheme>java CDS Sign.java
Prime p: 00 fc a6 82 ce 8e 12 ca ba 26 ef cc f7 11 0e 52 6d b0 78 b0 5e de cb cd
 1e b4 a2 08 f3 ae 16 17 ae 01 f3 5b 91 a4 7e 6d f6 34 13 c5 e1 2e d0 89 9b cd 1
3 2a cd 50 d9 91 51 bd c4 3e e7 37 59 2e 17


Generator g: 67 84 71 b2 7a 9c f4 4e e9 1a 49 c5 14 7d b1 a9 aa f2 44 f0 5a 43 4
d 64 86 93 1d 2d 14 27 1b 9e 35 03 0b 71 fd 73 da 17 90 69 b3 2e 29 35 63 0e 1c
20 62 35 4d 0d a2 0a 6c 41 6e 50 be 79 4c a4


Private Key X: 0e 19 65 16 85 85 a8 6e 97 70 48 35 a9 bb 41 fb c1 35 71 32


Public Key Y: 08 27 a4 a7 18 b2 4b 7b 55 2d d7 c4 0f 23 05 89 13 75 30 ca 95 19
fc 29 34 38 b5 8f 1b be eb d7 20 b5 ba b8 fa 7b 10 ce d8 5b 19 3a 4a 4b 3a 91 f4
 d6 98 23 c8 e3 93 0e 5c 93 f3 23 67 ca 84 94
X2= 4c 24 72 41 06 7b be 17 87 3c 9f 8d 72 00 e5 fd 94 28 c4 4e

Y2= 00 af a2 93 5b ac 23 cc 23 16 6c 24 47 d8 53 da c9 dc 38 bf 8f ca 8b 28 a2 9
6 f7 3f d3 cb b9 88 80 28 14 b4 58 bd cb e0 be bd 03 07 b8 6f c0 6b 71 3f ba cc
24 24 bf 36 99 d8 f1 5a 81 64 26 67 5f
K= 07 aa 80 2b c8 f1 b8 da 12 4f 90


S=137688412697287516934145706493085845330659633007583353350265545083825617244510
9248440422598136542970377579688429628311915393107669528938254247510623674349



User Private Key:: 1a 4a 11 c5 8a c9 46 45 40 30 b4 b4 ce 37 96 45 5b 81 29 2a 1
6 d1 27 b9 25 fb 13 3a 3b 70 f2 29 d1 fb 73 48 a0 ba 5d 82 eb 5a fa 6a 27 2a 91
e4 94 43 7b 39 c8 9e ce 53 04 da 61 5e ff 28 5f ed


User Public key Pair:: 00 af a2 93 5b ac 23 cc 23 16 6c 24 47 d8 53 da c9 dc 38
bf 8f ca 8b 28 a2 96 f7 3f d3 cb b9 88 80 28 14 b4 58 bd cb e0 be bd 03 07 b8 6f
 c0 6b 71 3f ba cc 24 24 bf 36 99 d8 f1 5a 81 64 26 67 5f


4c 24 72 41 06 7b be 17 87 3c 9f 8d 72 00 e5 fd 94 28 c4 4e


Signing Time::31 miliseconds


t:: 5f ac 7e ba d1 a4 61 15 0b 94 5d 3d 55 a5 02 12 3d d2 b5 0d

s:: 1a 4a 11 c5 8a c9 46 45 40 30 b4 b4 ce 37 96 45 5b 81 29 2a 16 d1 27 b9 25 f
b 13 3a 3b 70 f2 29 d1 fb 73 48 a0 ba 5d 82 eb 5a fa 6a 27 2a 91 e4 94 43 7b 39
c8 9e ce 53 04 da 61 5e ff 28 5f ed


Signature length is:: 328


1a 4a 11 c5 8a c9 46 45 40 30 b4 b4 ce 37 96 45 5b 81 29 2a 16 d1 27 b9 25 fb 13
 3a 3b 70 f2 29 d1 fb 73 48 a0 ba 5d 82 eb 5a fa 6a 27 2a 91 e4 94 43 7b 39 c8 9
e ce 53 04 da 61 5e ff 28 5f ed
```

```
t:: 5f ac 7e ba d1 a4 61 15 0b 94 5d 3d 55 a5 02 12 3d d2 b5 0d

s:: 1a 4a 11 c5 8a c9 46 45 40 30 b4 b4 ce 37 96 45 5b 81 29 2a 16 d1 27 b9 25 f
b 13 3a 3b 70 f2 29 d1 fb 73 48 a0 ba 5d 82 eb 5a fa 6a 27 2a 91 e4 94 43 7b 39
c8 9e ce 53 04 da 61 5e ff 28 5f ed


Signature length is:: 328


1a 4a 11 c5 8a c9 46 45 40 30 b4 b4 ce 37 96 45 5b 81 29 2a 16 d1 27 b9 25 fb 13
 3a 3b 70 f2 29 d1 fb 73 48 a0 ba 5d 82 eb 5a fa 6a 27 2a 91 e4 94 43 7b 39 c8 9
e ce 53 04 da 61 5e ff 28 5f ed


rhs:: 5f ac 7e ba d1 a4 61 15 0b 94 5d 3d 55 a5 02 12 3d d2 b5 0d


Verification Time::7 miliseconds
Signature Verified

C:\essential\Proj\project_modified_scheme>
```

Fig.4.1 Output of the Proposed Scheme

CHAPTER 5

SECURITY ANALYSIS

Security Analysis

We have done the security analysis for both our CDS scheme and the offline e-cash system which are described as follows.

5. 1 Security Analysis of Proposed CDS Scheme

In this section we will analyse the attacks based on CDS scheme and specify how our scheme is secured against them. Our scheme is entirely depend on the discrete logarithm problem (DLP) and discrete logarithmic assumption (DLA).

DEFINITION 1: *(DISCRETE LOGARITHM PROBLEM).* Let p be a large prime and g a generator in Z_p^* . Given (p, g, y)find the value x such that $y = g^x \ mod \ p$.

DEFINITION 2: *(DISCRETE LOGARITHM ASSUMPTION).* It is computationally infeasible to solve the discrete logarithmic problem i.e. it's a NP-Hard problem.

In our security analysis we consider two attacks, which are defined as follows:

DEFINITION 3: *(TYPE I ATTACK).*In this type of attack the adversary $A_I$who knows the public keys, try to determine the master private key of private key generator (PKG).

DEFINITION 4: *(TYPE II ATTACK).*In this attack the adversary A_{II}, who is a dishonest PKG knows the partial private key and public key tries to determine the user's key in user key generation.

<u>THEOREM 1-</u> *The proposed scheme is secure against TYPE I attack*

PROOF- In TYPE I attack the adversary A_I knows the public keys of the PKG. Now in order to get the master private key x, adversary A_I has to solve the DLP, i.e. $y = g^x \ mod \ p$, which is impossible as well as in feasible from the DLA. Hence our scheme is secured from TYPE I ATTACK.

THEOREM 2- *Proposed scheme is also secured against TYPE II ATTACK*

PROOF- in *TYPE II ATTACK* adversary A_{II} is a dishonest PKG, who knows the public key and partial private key in user key generation phase, and tries to find the user's private key. In proposed scheme user's private key is x_B, which was generated randomly from Z_q^*, and only can be known if the DLP can be solved for y_B. Hence it is secured against *TYPE II ATTACK.*

5.2 Security Analysis of E-Cash System

THEOREM 1- *if the blinding scheme is secure against forgery then the e-coin is also un-forgeable.*

Proof- If an adversary tries to forge an e-coin, he/she, must have to generate a valid blind signature of the Bank. Since DLA (Definition 2) says that DLP (Definition 1) is infeasible hence is our system un-forgeable.

THEOREM 2- *The anonymity of spender can be removed with the cooperation between the bank and TTP.*

Proof- In previously described e-cash system, we are always keeping records of the customer w.r.t the e-coin and is stored in linked to the customer with the bank and the TTP during the withdrawal protocol. Hence the bank can check the database and find the customer's ID to remove the anonymity.

The proposed scheme gives the facility to the user , as he/she can make anonymous payment with the merchant as the merchant cannot know the identification of a customer, he can only receive a coin from the user and verify the validity of the signature but cannot determine the customers identity . So the proposed scheme is withstand against Anonymity property.

CHAPTER 6

FUTURE SCOPE & CONCLUSION

6.1. FUTURE SCOPE

Our scheme can be used in the traditional e-cash system for the signature generation and verification, which will decrease the communication bandwidth and the signing and verification time. Here in this section we are proposing the framework for the e-cash system and all the detailed parameters that all the entities/parties are going to use. This framework was proposed by Popescu and Oros [5]. We have slightly modified their scheme and introduced our scheme in it.

6.2. CONCLUSION

In this thesis, we proposed a new modified Certificate less digital signature (CDS) scheme with improved signing and verification times and complexity. We also incorporated the proposed scheme in the fair off-line electronic cash system with anonymity revoking trustee. We also employed the customer tracing and the coin tracing to achieve all the features of an ideal e-cash system. This scheme confirms authenticity of the digitally signed document, anonymity of the signer and non-repudiation of the signature generation process. This scheme can also be applicable to many real life scenarios, such as, e-banking, online auction and electronic voting system.

REFERENCES

[1]Lein Harn, Jian Ren, Changlu Lin, ‘Design of DL-based certificate less digital signatures’, The Journal of Systems and Software 82 (2009) 789–793

[2] IITF principles, supra note 19, at 5.

[3] E.Brickell, P.Gemmel and D.Kravitz, ‘Trustee-based tracing extensions to anonymous cash and the making of anonymous change’, proceedings of The 6th ACM-SIAM,pp.457-466,1995.

[4] B. A. Fourazan, Debdeep Mukhopadhyay,’Cryptography and Network Security’, Tata McGraw Hill, 2nd edition,2010

[5] Constantin Popescu, Horea Orors, ‘A fair off-line electronic cash system with anonymity revoking trustee’.Proceedings of the International Conference on Theory and Application of Mathematics and Informatics-ICTAMI 2004, Thessaloniki, Greece

[6] Lee, Chang, ‘Strong designated verifier signature scheme’, Computer Standard and Interface, 31, 2009

[7] Al-Riyami, S., Paterson, K., 2003. ’Certificateless public key cryptography’. Advances in Cryptology – AsiaCrypt, LNCS, vol. 2894. Springer-Verlag, pp. 452–473.

[8] Bellare, M., Namprempre, C., Neven, G., 2004. ‘Security proofs for identity-based identification and signature schemes’. Advances in Cryptology – EuroCrypt’04, LNCS, vol. 3027. Springer-Verlag, pp. 268–286.

[9] Mafruz Zaman Ashrafi, ‘Privacy-preserving e-payments using one-time payment details,, Journal of Systems and software 31 (2009) 321–328

[10] S. Brands, ‘Untraceable off-line cash in wallet with observers', Advances in Cryptology-CRYPTO’93, Lecture Notes in Computer Science, Vol. 773, Springer-Verlag, pp. 302-318,1993.

[11] J. Camenisch, J. Piveteau, M. Stadler, ‘An efficient payment system , protecting privacy’, proceedings of ESORICS’94, Lecture Notes in computer science, Vol. 875, Springer –Verlag, pp. 207-215, 1994.

[12] Cheng-Chi Lee , Min-Shiang Hwang , Wei-Pang Yang, ‘A new blind signature based on the discrete logarithm problem for untraceability’

[13] Cha, J., Cheon, J.H., 2003. ‘An identity-based signature from gap Diffie-Hellman Groups’. Public Key Cryptography – PKC’03, LNCS, vol. 2567. Springer-Verlag, pp.18–30.

[14] Chen, X., Zhang, F., Kim, K., 2003. A new ID-based group signature scheme from Bilinear pairings. WISA’03, LNCS, vol. 2908. Springer-Verlag, pp. 585–592