TIMESTAMPED DIGITAL SIGNATURE SCHEME WITH MESSAGE RECOVERY & ITS APPLICATION IN E-CASH SYSTEM

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

Bachelor of Technology

In

Computer Science & Engineering

By

P B Sai pavan kumar (111CS0132)

&

Kodavatikanti Hanok (111CS0172)

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

DECLARATION OF AUTHORSHIP

We, Sai pavan Pothuri and Hanok kodavatikanti, declare that thesis titled “Timestamped Digital Signature Scheme with Message Recovery & its Application in E-cash System” and the work presented in it are our own work, we confirm that :

- This work was done wholly or mainly while in candidature for research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work
- We have acknowledged all main sources of help
- Where the thesis is based on work done by our self jointly with other, we have made clear exactly what was done by others and what we have contributed our self.

P B Sai pavan kumar
(111CS0132)
B. Tech.
Computer Science & Engg.
NIT Rourkela

K Hanok
(111CS0172)
B. Tech.
Computer Science & Engg.
NIT Rourkela

ACKNOWLEDGEMENT

We are indebted to our guide **Dr. 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.

P B Sai pavan kumar
(111CS0132)
B. Tech.
Computer Science & Engg.
NIT Rourkela

K Hanok
(111CS0172)
B. Tech.
Computer Science & Engg.
NIT Rourkela

ABSTRACT

We propose a Timestamped signature scheme which can be verified universally using signer's public parameters. A trusted third party, the Timestamping System provides timestamp to a signature without even knowing the content of the document. The proposed scheme can withstand active attacks, such as forgery attack and chosen cipher text attack. It also provides the message recovery feature, i.e., from the timestamped signature, the message can be recovered by the receiver. Hence, the message need not be sent with the signature. The suggested scheme do not require any hash function and there by reduces the verification cost as compared to existing schemes at the expense of marginal increase in signature generation cost. Further, the scheme is more secured as its security lies in solving three computationally hard assumptions Performance analysis of both the schemes has been carried out in details. We applied the Time-stamped signature scheme with Message recovery in E-cash system.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION:

For any commercial electronic applications authenticity of any document plays a crucial role. A Digital signature should provide the authenticity, data integrity and non-repudation. In the Zuhua Shao's scheme the validity of the signed document can be verified by any recipient. I this scheme the signer authenticates the digest with the private key of his own and the recipient can verify the authenticity with the public key of the corresponding private key. Without the knowledge of the private key of the signer, the signatures cannot be forged. And without he time stamp, the document cannot be trusted if the key is lost or compromised. So a time stamp service should be established to provide timestamps for every document that a signer signs [2,3].

A timestamp ensures non-repudiation. Timestamps are provided by trusted third parties, known as Time Stamping Systems (TSS). It provides integrity, authentication, and non-repudiation of document. An authentic timestamp on a signed message ensures that even if the secret key of the signer is leaked or compromised, the authenticity and non-repudiation features remain intact. Time stamping is compulsory in many applications such as e-cash systems and electronic commerce. In all these applications, non-repudiation of a signed document is very much important. For instance, in e-cash systems, electronic coins should be associated with a date and time where they were signed by concerned parties. Hence, time stamping\ is widely recognized as a methodology to certify that a message was modified or signed at a certain point of time. In long term preservation of digital signatures timestamps are widely used. Many digital timestamping systems exist, which are based on digital signatures and collision free hash functions [2,3].

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 [2,3].

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 [2,3].

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. There are two keys one is *public key* and the other one is *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 [2,3].

- RSA Public Key Cryptosystem
- Elgamal Public Key Cryptosystem

2.1 Digital Signature

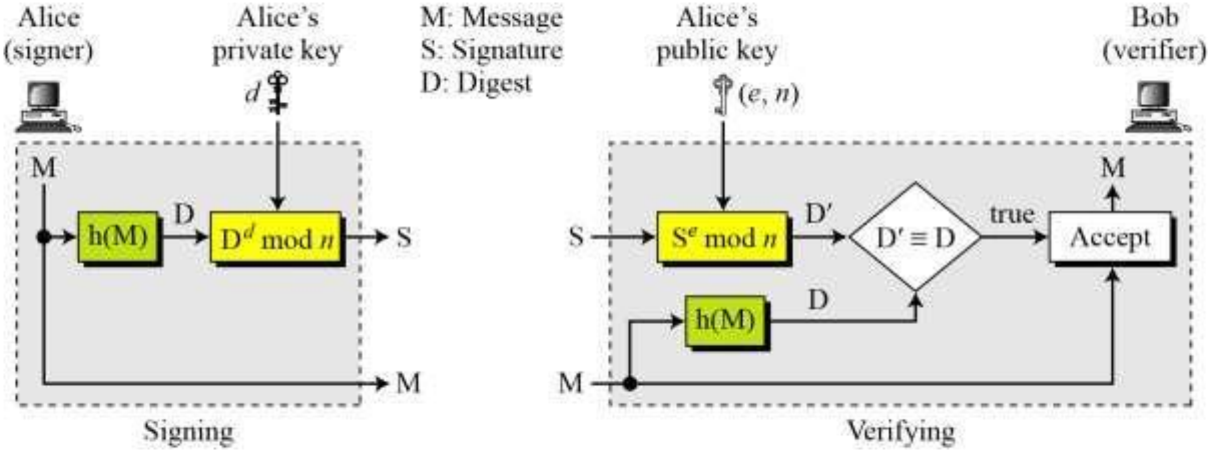
A conventional signature is actually a part of the document which is included in the document. But in case of Digital signature it is sent along with the document. The sender sends the Message along with the signature where both are separate documents. Then the recipient receives both the message and signature. And he authenticates the document.

Several digital signature schemes have evolved during the last few decades. In this section, we discuss some of the widely used signature schemes [2,3].

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 [2,3].

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

$S_1 = e_1^{r}$ mod p and $S_2 = (M - d * S_1) *r^{-1}$ mod (p-1), where S_1 and S_2 are the two signatures.

The sender sends M, S_1, *and S2* to the receiver. To verify the message, the verifier calculates [6].

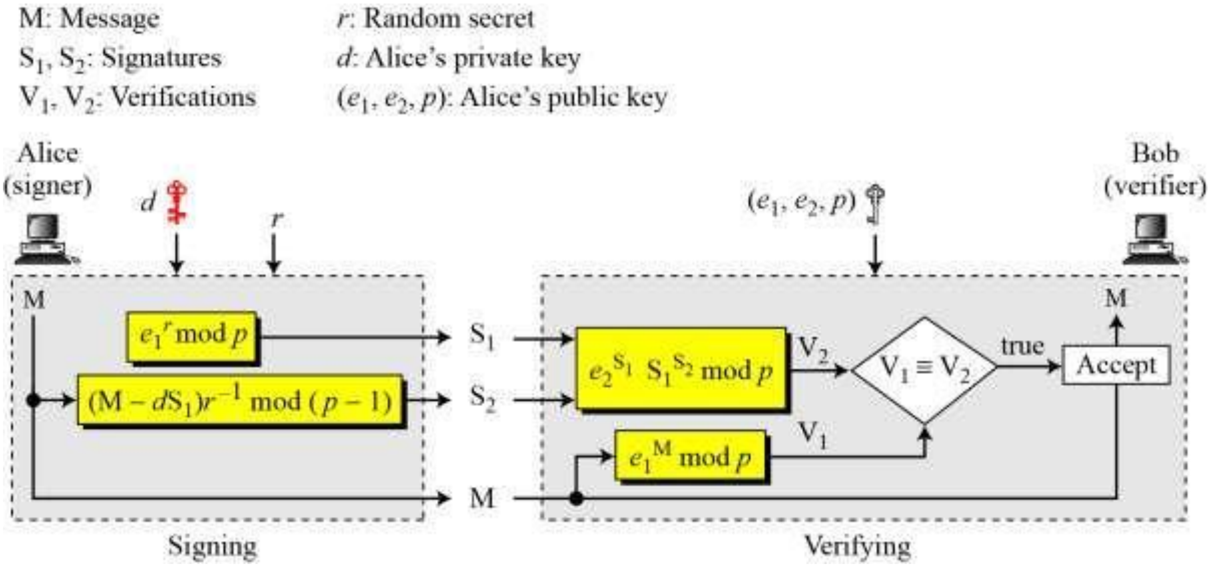


Fig.2.2 ElGamal digital signature scheme

$V_1 = e_1^{M}$ mod p and $V_2 = e_2^{S_1} * S_1^{S_2}$ mod 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]:

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].

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 [2].

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 TSS with message Recovery.

2.5. Timestamped signatures with Message Recovery

In the proposed scheme only the signature is sent through the channel. And the message can be recovered from the signature itself by using signer's public parameters. The Original message is recovered from the signature itself. The DSS with message recovery consists of two parties; one is the signer and the other is a trusted third party known as Time stamping System (TSS). The signer has to generate signature for a message *m* with an authenticated timestamp issued from the TSS. This scheme consists of following three phases, namely, setup phase, signature verification and signature generation [7,8].

Examples: RSA, Rabin, Nyberg-Rueppel

CHAPTER 3

ZUHUA SHAO'S MESSAGE HIDDEN TIMESTAMPED SIGNATURE

3.1 Review

Basically there are three parties in the Existing scheme:

(1) With the authenticated timestamps the signer calculates the signature and signs the document;

(2) The TSS will add timestamp to the signature; and

(3) The verifier will check the validity of timestamped signatures.

3.1.1 System parameters:

For a timestamped signature scheme, the TSS chooses a large prime p and a generator g ∈ Z^{*}_{p} with order q, where q is a prime divisor of $p – 1$, *and a hash function* h *( ).* Here p, g and h ( ). Here p, g *and* h *()* are public system parameters which are authentically known to all users. The TSS chooses his private key $x_N \in Z^{*}_{q}$ and computes his public key $y_N = g^{x}{}_{N}$ (mod p). Similarly, the signer chooses a random number $x_S \in Z^{*}_{q}$ as his own private key and computes $y_S = g^{x}{}_{S}$ (mod p) as his public key [1].

3.1.2 Signature Generation & Signature verification:

First, the signer computes signature for the message M. The signer chooses a random number $k1 \in Z_q^{*}$, and computes $r1 = g^{k1}$ (mod p), $e1 = h\ (M, r1)$ and $s_1 = e_1x_S + k_1$ (mod q). Then signer will send the tuple as the signature for the message M to a recipient.

From the relation The recipient computes $r_1=g^{s}{}_{1}y_S{}^{-e1}$ *(mod p)* and checks $e_1=h\ (M, r_1)$.

The recipient sends (s_1,e_1) to a TSS inorder to generate the Timestamped signature for the message M.

The TSS selects a random number $k_2 \in Z_q{}^*$, and computes $r_2=g^{k}{}_{2}\ (mod\ p)$, $e_2=h\ (t, e_1, r_2)$ where t is the timestamp. Then TSS sends (t, s, r_1, e_2) to the recipient.

Finally the recipient receives the timestamped signature $(s, r_{1,}\ e_2)$ of the message and timestamp t, which can be verified by checking the following verification equation.

$e=h(t, h(M,r_1), g^{S}\ (\ (y_S\, y_N\ )^{h(M,r_1)}\ r\ )^{e}{}_{2}\ (\ mod\ p\ )).$

Similarly the timestamped signature $(s, r_{1,}\ e_2)$ of the message M and the time information t is always to be accepted.

From the relation e_1 and e_2 it is clear that for signing the timestamp t is needed for non-repudiation. From this receiver can verify the signature whether it is not valid and can easil predict the time of signing [1].

3.2 Proposed Timestamped Digital Signature Scheme with Message Recovery:

The proposed scheme consists of two parties; one is the signer and the other is a trusted third party known as Time Stamping System (TSS). The signer has to generate signature for a message *m* with an authentic timestamp issued from the TSS. This scheme consists of following three phases:

- Setup phase
- Signature Generation
- Signature Verification

3.2.1 Setup:

The signer selects an integer *n* which is equals to the multiplication of big prime numbers *p* and *q* where $p=2p'f+1$ and $q=2q'f+1$. Here, *f*, p´ and *q´* are all prime numbers. Then it chooses a generator g of order f both modulo p and q. After that, it selects e which is co-prime with both p-1 and q-1 and d is computed using $ed \equiv 1 \mod \Phi(n)$, where $\Phi(n)$ is Euler's Totient function.

Signer chooses its secret key x in random from Z_{n^*} and computes public key as $Y = g^x$ mod n. Finally, the signer publishes the system's public parameters(Y, g, n, e) and keeps d, p, q, f, x as secret.

3.2.2 Signature Generation:

Step 1: First, the signer requests the TSS for a timestamp. The TSS acknowledges the request by sending t in a secure channel to the signer. Here t is the current date and time.

Step 2: After receiving the timestamp t, the signer randomly chooses $u \in Z_n^*$ and computes S and r_1 as follows.

$$S = Y^d \bmod n \qquad (10)$$

$$r_1 = (S + mg^u) \bmod n \qquad (11)$$

The signer then sends (S, r_1, t) to TSS.

Step 3: After receiving (S, r_1, t), the TSS first checks the validity of t. Then it proceeds to *Step 4* if the condition satisfies. Otherwise it rejects the signature (S, r_1, t) and asks the signer to resend the data.

$$S^e = Y \bmod n \qquad (12)$$

Step 4: The TTS chooses an integer $k \in Z_n^*$ and computes the values of r_2 and r as follows.

$$r_2 = (k - r_1 - t) \bmod n \qquad (13)$$

$$r = (r_1 - S)g^{-k} \bmod n \qquad (14)$$

Then it sends (r, r_1, r_2, t) to the signer.

Step 5: After receiving (r, r_1, r_2, t), the signer checks it with his own (r_1, t) as derived in *Step 1*. If they match, then he accepts the received data and computes ℓ from the following linear equation.

$$S + \ell \equiv x^{-1}(r_2 - u + r_1) \bmod n \quad (15)$$

Timestamped signature of the message m is (t, S, r, ℓ).

3.2.3 Signature Verification:

Any verifier or receiver can verify the signature (t, S, r, ℓ) of message m by checking the following condition given as,

$$S^e = Y \bmod n \qquad (16)$$

The message m can be recovered as,

$$m = rY^{S+\ell} g^t \bmod \mathrm{n} \qquad (17)$$

The message is indeed the original one as,

$$\begin{aligned} rY^{S+\ell} g^t \bmod \mathrm{n} &= (r_1 - S)g^{-k} Y^{(x^{-1}(r_2-u+r_1))} g^t \bmod \mathrm{n} \\ &= (r_1 - S)g^{-k} g^{x(x^{-1}(r_2-u+r_1))} g^t \bmod \mathrm{n} \\ &= (mg^u)g^{-k} g^{(r_2-u+r_1)} g^t \bmod \mathrm{n} \\ &= mg^{(u-k+r_2-u+r_1+t)} \bmod \mathrm{n} \\ &= mg^{(k-r_1-t-k+t+r_1)} \bmod \mathrm{n} \\ &= m \bmod n \end{aligned}$$

CHAPTER 4

PERFORMANCE EVALUATION

4.1 PERFORMANCE STUDY

The complexity of any digital signature scheme depends on four operations, namely, multiplication, exponentiation, hash functions and inverse operation. The performance of the proposed scheme can be studied by comparing it with some existing schemes [1, 2]. The results of performance are tabulated in Table 1. The time for computing the modular addition and subtraction computations are neglected. The performance is analyzed with the following notations.

- T_E is modular exponentiation time
- T_M is modular multiplication time
- T_I is modular inverse operation time.
- T_H is hash functions performance time.

It may be observed that the executional cost of signature verification phase of the proposed scheme is pretty lower than Shao's scheme as well as Sun et. al's scheme. The proposed scheme is does not have any hash function unlike Shao's scheme. The cost of signature generation of the proposed scheme is higher than that of both the schemes due to more number of inverse operations. However, the proposed scheme is more secured as compared to other two schemes.

4.2 Comparison with Existing Scheme

Table I: Execution comparison

Phases	**Shao's Scheme [1]**	**Sun's scheme [2]**	**Proposed Scheme**
Signature generation	$4T_E + 4T_M + 1T_I + 3T_H$	$2T_E + 6T_M$	$4T_E + 4T_M + 3T_I$
Signature verification	$3T_E + 3T_M + 3T_H$	$4T_E + 3T_M$	$2T_E + 2T_M$

- Time required for Exponential Operation

- Time required for Multiplication operation

- Time required for Hash operation

- Time required for Inverse operation

```
48
output of the TSS
3621
2015-03-09
 06:05:33
14432245684896487005991660720
87174119458164634120
5971
verified signature
1443224568489648700599166072087174119458164634120
```

Fig.4.1 Output of the Proposed Scheme

CHAPTER 5

SECURITY ANALYSIS

Security Analysis

We have done the security analysis for both our TSS with message recovery and the offline e-cash system which are described as follows.

5. 1 Security Analysis of Proposed TSS Scheme with Message Recovery

In this section we will analyze the attacks based on TSS scheme and specify how our scheme is secured against them. Our scheme is entirely depend on the discrete logarithm problem (DLP) and Integer factorization problem (IFP).

DEFINITION 1: *(DISCRETE LOGARITHM PROBLEM).* The discrete logarithm problem is defined as: given a generator *g of* a group *G*, and an element *h* of *G*, we have to find the DL to the base *g* of *h* in the group *G*. DLP is not generally hard. The hardness of DL relies on the groups.

DEFINITION 2: *(TYPE I ATTACK).* It is difficult to obtain the secret key *x* from the public key *Y*. An attacker who has access to public key Y has to solve the equations $Y = g^x \bmod n$, which is clearly a DLP. Also if he needs to factor *n*, then he has to solve the IFP, as $n = pq$. To obtain the values of *p* and *q*, he needs to factor *p'* and *q'*, which is computationally infeasible.

DEFINITION 3: *(TYPE II ATTACK).* Proposed scheme can withstand parameter reduction attack. Equation (17) can be rewritten as follows.

$$m = r.\ Y^{t'}\ g^{t} \mod n,$$

Where $t' = S + \ell$ and the parameters in the above expression cannot be reduced further. Hence the proposed scheme can combat parameter reduction attack.

DEFINITION 4 An intruder cannot fake a valid timestamp**.** If an intruder attempts to forge a valid signature, he needs secret keys of both signer and TSS, whose security lies in the difficulty of solving DLP and IFP. Furthermore, if TSS is dishonest, then also he cannot fake an authenticated timestamp without collaboration of signer. Also the TSS can not acquire any information of message from the timestamp.

CHAPTER 6

E-cash System

6.1 Introduction

An e-cash system can be classified as offline or online. In offline e-cash system customer pays merchant without help of bank means there is no role of bank during transaction but during deposit merchant will deposit the e-coin in the bank. When customer pays merchant, bank should be online means presence of bank is needed during payment. In online e-cash system all participants should be online. An e-cash systems replace the traditional system. Payments less than "one" is possible through e-cash systems. E-cash systems are used in online banking systems; they are also used for online shopping and other purposes. There are many features those can be achieved by e-cash System such as [9].

***Transferability*:** Electronic coins will be distributed among people whether the transactions are online or offline.

***Portability*:** The security of e-coin does not depend upon any physical location. Using computer networks we can transfer into storage devices and vice versa.

***Off-line Payment*:** Offline payment means there is no need of bank presence during transaction between the customer and the merchant

***Unforgeability*:** Unforgeability means only Bank can make coins which is an authorized party.

***Anonymity*:** The spender of the e-coin must stay unknown. On the off chance that the coin is spend truly, neither the beneficiary nor the bank can distinguish the spender.

***Unreusability*:** The e-coin ought not be replicated and reused. At that point we need to Minimize the dangers for forgery and should create a good authentication system.

Divisibility: E-coin can be divided into littler sums [9].

6.2 Application of proposed scheme in offline e-cash

Introduction

Customer: Customer buys goods or stuffs and gives e-coin to merchant rather giving cash or cheque.

Merchant: Merchant takes e-coin from the customer and deposits into the bank where the purchase amount will be deducted from customer's account and credited to merchant's account.

Bank: Bank tracks all the transaction between merchant and customer, all the details of customer and merchant.

Timestamping System: TSS is a third party, it publishes the public keys and private keys for digital signature and cryptography.

6.2.1 System Setup

In this protocol, bank and TSS create the public keys and private keys. They will announce their public keys publicly. These are the following steps:

Step 1: Bank randomly chooses 3 prime numbers $p0$, $q0$ and f such that $p = (2 * p0 * f + 1)$ and $q = (2 * q0 * f + 1)$ are primes.

Step 2: Chooses e which is co-prime with both $p - 1$ and $q - 1$.

Step 3: Computes $n = p * q$, $\varphi(\mathrm{n})=(p - 1) * (q - 1)$ and $d=e-1 \bmod \varphi(\mathrm{n})$.

Step 4: Chooses g as a generator of n of order f.

Step 5: Clearing house chooses x in $Z\varphi(n)*$ and computes $Y = gx \bmod n$.

Step 6: Publishes Public key(Y, g, n, e) , Private key (d, p, q, f, x).

Step 7: Finally Digital signature is calculated using the signer's private key and it is verified by the receiver using signer's public key.

6.2.2 Withdrawal Protocol:

Step1:

Customer requests the TSS for time t, The TSS acknowledges the request by sending ’t’ in a secure channel

Step2: After receiving the timestamp t, the customer randomly chooses $u \in Z_n^*$

And computes s and as follows

$$S = Y^d \bmod n$$

$$m = H(M)$$

M = Amount needed

Then the customer sends (S,m) to bank.

Step3: Bank receives the (S,m) and verifies the Signature with

$$S^e = Y \bmod n$$

Bank produces a coin containing coid ID, coin Amount, expiry (coin information) and sends it to bank.

Step4: Customer receives the message m = H (CI) with the Bank timestamp signature.

$S = Y^d \bmod n$

Customer gets (S,m) and checks the validity with the equation

$$S^e = Y \bmod n$$

6.2.3 Payment Protocol:

Step1:

Customer first requests the TSS for time.

Step2:

After receiving the timestamp t, the Customer randomly chooses $u \in Z_n^*$

And computes S and r_1 as follows

$$S = Y^d \text{ mod n.}$$

$$r_1 = (S + mg^u) \text{ mod n}$$

Customer then sends (S, r_1, t) to TSS.

Step3:

After receiving (S, r_1, t) the TSS first checks the validity of t. Then it proceeds to step 4 if the condition satisfies

$$S^e = Y \text{ mod n}$$

Step4: The TSS chooses an integer k from Z^*_n and computes the values of r_2 and r as follows.

$$r_2 = (k-r_1-t) \text{ mod n.}$$

$$r = (r_1-s)\, g^{-k} \text{ mod n.}$$

Step5: After receiving (r,$r_{1,}$ r_2, t) the signer checks it with his own (r_1,t) as derived to step 1. If they match then it accepts the received data and computes l from the equation

$$S + l = (r_2-u+r_1) \text{ mod n}$$

The time stamped signature of coin n is (t, S, r, l)

Step 6:

The (t,S,r,l) is received by the Merchant and verifies with

$$S^e = Y \text{ mod n.}$$

And coin info with

$$m = rg^{Y+l}g^t \text{ mod n}$$

6.2.4 Deposit Protocol:

Merchant along with the signature of it sends (t, S_c, S_M, r, l) to Bank.

Bank verifies the Merchant with

$$S_M^e = Y \text{ mod n}$$

Also checks for the double spending of the coin.

CHAPTER 7

CONCLUSION & FUTURESCOPE

7.1. CONCLUSION & FUTURE SCOPE

This Thesis presents a distinct Time stamped signature scheme (TSS) based on computationally hard problems. This scheme consists of less computational overhead and supports message recovery feature. Moreover, this scheme is resistant to forgery attack and parameter reduction attack. The power of the trusted third party TTS is limited. This scheme also supports message recovery, hence reduces the communicational overhead. This signature scheme confirms non-repudiation property. This scheme is secure, hence can be applicable to areas such as e-cash, e-bidding and e-commerce applications. 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. We have slightly modified their scheme and introduced our scheme in it. In our new E-cash scheme, we propose a new off-line E-cash system which maintains the security features like anonymity, double spending, unforgeability. It is also immune to various CCA attacks and can withstand them; it implements the tracing protocol. We use three different protocols: account opening, payment and deposit protocol for the development of the e-cash system which can be used for secure payment across internet and other networks. An E-cash system which incorporates all the security features in it and helps in the secure payments to the merchants. It also takes care of the various attacks to which the e-cash is vulnerable and tries to resist them; In a nutshell, this e-cash scheme facilitates payments to the merchants and helps in exchange of merchandise also by implementing various protocols for secure communication between the merchants and the customers as well as between the merchants and the Bank. In future research can be done on our scheme to lower its computation cost and communication overhead. Also research can be done to incorporate timestamping feature to some of the highly proved secured signcryption schemes which can be applicable to highly security sensitive application like e-bidding, e-voting, e-transactions etc.

Dissemination of our Work

S.Mohanty, Sai Pavan Kumar P B,Hanok K," A Timestamped Signature Scheme with Message Recovery", International Conference on High Performance Computing and Applications (ICHPCA) 2014,pages 1-4,DOI:10.10.1109/ICHPCA2014.7045299

REFERENCES

[1] Z. Shao, “Security of the design of timestamped signature”, Journal of Computer and System Sciences, 72, pp. 690-705 (2006).

[2] Behrouz A. Forouzan. Cryptography and Network Security . Tata McGraw-Hill, 2007.

[3] William Stallings. Cryptography and Network Security: Principles and Practices. Prentice Hall Inc., 1999.

[4] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22(5):644-654, 1976.

[5] H.M. Sun, B.-C. Chen, H.-T. Yeh, “On the design of timestamped signatures”, Journal of Computer and System Sciences, 68, pp. 598-610 (2004).

[6] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithm," IEEE Trans. Inform. Theory, vol. IT-31, no. 4, pp. 469-472, 1985.

[7] J. Benaloh, M. de Mare, “Efficient broadcast timestamping”, Clarkson University, Department of Mathematics and Computer Science, TR 91-1, August 1991.

[8] S. Haber, W.S. Stornetta, “How to timestamp a digital document”, J. Cryptology 3 (2) (1991) 99–111.

[9] Ziba Eslami, Mehdi Talebi A new untraceable offline electronic cash system Department of Computer Science,Shahid Behesti University, G.C., Tehran, Iran,2011

[10] D. Bayer, S. Haber, W.S. Stornetta, “Improving the efficiency and reliability of digital timestamping”, in: R.M. Capocelli (Ed.), Sequences ’91: Methods in Communication, Security, and Computer Science, Springer-Verlag, Berlin, 1992, pp. 329–334.

[11] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, Internet X. 509, Public key infrastructure timestamp protocol, (TSP), RFC 3161, August, 2001.

[12] X.T. Fu, C.X. Xu and G.Z. Xiao, "Forgery Attacks on Chang et al.'s Signature Scheme with Message Recovery, “Cryptology ePrint Archive, Report 2004/236, 2004. http://eprint.iacr.org/.

[13] C.C. Chang and YF. Chang, "Signing a Digital Signature without using one-way Hash Functions and Message Redundancy Schemes," IEEE Commun. Lett., vol. 8, no. 8, pp. 485-487, Aug. 2004.

[14] A. Bonnecaze, P. Liardet, A. Gabillon and K. Blibech, “A distributed Time Stamping Scheme”, IEEE SITIS conference, December 2005.

[15] X. Hou and C. Tan, “A new electronic cash model”, Proc. IEEE International Conference on Information Technology: Coding and Computing (ITCC’05), pages 374-379, 2005.

[16] S. Mohanty and B. Majhi, “A digital signature scheme with message recovery and without one-way hash function”, International Conference on Advances in Computer Engineering PP. 265 – 267, 2010

[17] S. Mohanty and B. Majhi, and SK Baral, “A novel timestamped signature scheme based upon DLP”, 1st International Conference on Recent Advances in Information Technology, RAIT-2012 PP. 6 – 10, 2012