Time Stamped Proxy Blind Signature Scheme With Proxy Revocation Based on Discrete Logarithm Problem

Suryakanta Panda

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

Time Stamped Proxy Blind Signature Scheme With Proxy Revocation Based on Discrete Logarithm Problem

Dissertation submitted in

May 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Suryakanta Panda

(Roll 211CS1273)

under the supervision of

Prof. Ramesh Kumar Mohapatra

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, Odisha, India

Dedicated to

Netaji Subhas Chandra Bose

and

Lal Bahadur Shastri

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

May 2013

Certificate

This is to certify that the work in the thesis entitled ***Time Stamped Proxy Blind Signature Scheme with Proxy Revocation Based on Discrete Logarithm Problem*** submitted by **Suryakanta Panda** is a record of an original 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 with the specialization of Computer Science in the department of Computer Science and Engineering, National Institute of Technology, Rourkela. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere.

Ramesh Kumar Mohapatra
Assistant Professor
Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Odisha-769008

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Ramesh Kumar Mohapatra, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Cryptography and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

My heartfelt thanks also to Santosh Kumar Sahu, Ph.D Scholar, Dept.of CSE, for his encouragement and support.

I am really thankful to my all friends. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, and their invaluable time, I am truly indebted.

My thanks and apologies to those whom I have inadvertently missed out.

Finally, I thank God for everything.

Suryakanta Panda

Abstract

Proxy blind signature combines both the properties of blind signature and proxy signature. In a proxy blind signature scheme, the proxy signer is allowed to generate a blind signature on behalf of the original signer. It is a protocol played by three parties in which a user obtains a proxy signer's signature for a desired message and the proxy signer learns nothing about the message. During the verification of a proxy blind signature scheme, the verifier cannot get whether signing is within the delegation period or after delegation period. In this thesis a time stamped proxy blind signature scheme with proxy revocation is proposed which records the time stamp during the proxy signing phase and satisfies all the security properties of proxy blind signature i.e distinguishability, nonrepudiation, unforgeability, verifiability, identifiability, unlinkability, prevention of misuse. In a proxy revocation scheme, the original signer can terminate the delegation power of a proxy signer before the completion of delegation period. Proxy blind signature has wide applications in real life scenarios, such as, e-cash, e-voting and e-commerece applications.

***Keywords*:** Proxy Blind Signature, Proxy Signature, Blind Signature, Proxy Revocation, DLP

Contents

Chapter 1

Introduction

A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. In short, it has the same function as that of a handwritten signature. Digital signatures provide even more security than their handwritten counterparts. 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 nonrepudiation) and that the message was not altered in transit (integrity). A digital signature scheme typically consists of three algorithms:

1. A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.

2. A signing algorithm that, given a message and a private key, produces a signature.

3. A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.

But the problems of digital signature come when the signer got the details about the user during transactions. A tremendous amount of data about a user's habits, affiliations, and lifestyle whereabouts can be captured by the signer in electronic form. This breaches the privacy of the person in concern. Organizations

now have massive amounts of data, threatening the user's security. A digital signature reveals the identity of the user in any transaction whereas a blind signature protects the user's privacy.

1.1 Blind Signature

Blind signature as introduced by David Chaum [1] is a form of digital signature in which the content of a message is blinded before it is signed. It allows a user to acquire a signature from the signer without revealing the message content for personal privacy. The resulting blind signature can be publicly verified against the original, unblinded message in the manner of a regular digital signature. In a blind signature scheme, the signer cannot link the relationship between the blind message and the signature of the chosen message. Blind signatures are typically employed in privacy-related protocols where the signer and message author are different parties [1,5]. Blind signature schemes have applications where the sender A (the customer) does not want the signer B (the bank) to be capable of associating a postiori message m and a signature $S_{Blind}(m)$ to a specific instance of the protocol. This may be important in electronic cash applications where a message m might represent a monetary value that A can spend. When m and $S_{Blind}(m)$ are represented to B for payment, B is unable to deduce which party was originally given the signed value. This allows A to remain anonymous so that spending patterns cannot be monitored.

1.2 Proxy Signature

A proxy signature protocol introduced by Mambo et al. [2], allows a designated person, called a proxy signer, to sign on behalf of an original signer, in case of saying, temporal absence, lack of time or computing power, etc. When a receiver verifies a proxy signature, he verifies the signature itself and original signer's delegation together. The basic methodology of proxy signature is that the original signer creates a signature on delegation information (ID of the proxy signer, or any

warrant information) and gives it secretly to the proxy signer, and then the proxy signer uses it as a proxy private key or uses it to generate a proxy private key. Because the proxy key pair is generated from the original signer's signature on delegation information, any verifier can check the original signer's agreement from a proxy signature [4, 15]. Once the proxy signer creates a valid proxy signature of the original signer, the proxy signer cannot repudiate his signature creation against anyone, and the original signer cannot deny that he delegates his signing power to the proxy signer.

Delegations of various kinds are very common in society. Delegation of signing power is one of them. Based on the different types of delegation Mambo et al. [2] classified proxy signature schemes into full delegation, partial delegation, delegation by warrant.

1.2.1 Full Delegation

In the full delegation, a proxy signer is given the same secret s that an original signer has, so that she can create the same signature as original signer creates. Obviously, when the proxy signer deliberately signs a document unfavorable for the original signer, her mischievous action is not detected because the signature created by the proxy signer is indistinguishable from the signatures created by the original signer.

1.2.2 Partial Delegation

In the partial delegation, a new secret d is created from s, which follows the modification of a verification equation, and d is given to a proxy signer in a secure way. The created signature is checked by the modified equation, but not by the original equation. That implies a signature created by the proxy signer is distinguishable from a signature created by the original signer, and the original signer, who has found a signed document with the content unfavorable for him, can distinguish his ordinary signature from a proxy signature for partial delegation. A proxy signature for each proxy is distinct. In this delegation, only the public

key of the original signer is required for the verification.

Two common classifications [4, 7, 8] are:

Proxy unprotected proxy signature

Besides proxy signer, original signer can create a valid proxy signature. But no third party, not designated as proxy signer, can create a valid proxy signature.

Proxy protected proxy signature

Only the proxy signer can able to create a valid proxy signature. No one other than proxy signer, not even the original signer can create a valid proxy signature.

1.2.3 Delegation by Warrant

A warrant is a certificate composed of a message part that the proxy signer is authorized to sign and a public key which ensures the involvement of the original signer.

There are two types of schemes for this purpose [2]:

Delegate Proxy

In delegate proxy, original signer, Alice, signs a warrant and declares Bob as designated proxy signer, under her secret key by an ordinary signature scheme. The warrant so created is given to Bob. Now Bob when wants to sign a message on behalf of Alice, he simply signs the message by his own key and combines the warrant with the message. Warrant is only identity which differentiate between Bob's normal signature and proxy signature.

Bearer Proxy

In bearer proxy, an original signer computes a proxy secret key and its corresponding public key. Original signer signs a warrant, composed of a condition

of authorization and newly generated public key. The secret key is given to proxy signer in a secure way.

Proxy signature schemes can be constructed for each of these delegation types. The partial delegation, and the delegation by warrant are more secure than the full delegation. The partial delegation has a computational advantage over the schemes with warrant. On the other hand delegation by warrant can be implemented by ordinary signature schemes without any modification, and it is appropriate for restricting the documents to be signed.

1.3 Proxy Blind Signature

The proxy signature and blind signature have respective advantages. In some real situations, we need to inherit the merits of both proxy and blind signatures. The first proxy blind signature was proposed by Lin et al. [6] in 2000. Proxy blind signature scheme is a digital signature scheme that combines the properties of both proxy signature and blind signature. In a proxy blind signature scheme, the proxy signer is allowed to generate a blind signature on behalf of the original signer. In the proxy signature scheme, the proxy signer knows the original message m, but in the proxy blind signature scheme, the proxy signer does not know the m. Proxy blind signature scheme is a protocol played by two parties in which a user obtains a proxy signer's signature for a desired message and the proxy signer learns nothing about the message [5-13,18,19].

1.4 Security Requirements of Proxy Blind Signature

The proxy blind signature satisfies the security properties of both the blind signature and the proxy signature, such signature is suitable for many applications where the users privacy and proxy signature are required.

1.4.1 Distinguishability

The proxy signature must be distinguishable from the normal signature.

1.4.2 Nonrepudiation

Neither the original signer nor the proxy signer must be able to sign in place of other party. They cannot deny their signatures against anyone.

1.4.3 Unforgeability

Only a designated proxy signer can create a valid proxy signature for the original signer. (Even the original signer cannot do it).

1.4.4 Verifiability

The receiver of the signature should be able to verify the proxy signature in a similar way to the original signer.

1.4.5 Identifiability

Anyone can determine the identity of the corresponding proxy signer from a proxy signature.

1.4.6 Prevention of Misuse

It should be confident that proxy key pair should be used only for creating proxy signature, which conforms to delegation information. In case of any misuse of proxy key pair, the responsibility of proxy signer should be determined explicitly.

1.4.7 Unlinkability

When the signature is verified, the signer knows neither the message nor the signature associated with the signature scheme.

1.5 Motivation

The motivation for this project came from the growing need for a proxy blind signature scheme which can assure maximum possible security from the existing schemes. In cases where the proxy signer abuses her/his delegated rights, the original signer needs to revoke the proxy signer's signing capability. Proxy signer can make fool the verifier by signing the message after the delegation period is over.

1.6 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 describes about the literature surveys that have been done during the research work. The Tan et al. scheme [7], Lal et al. scheme [8], Xue et al. scheme [12] and Yang et al. scheme [34] are discussed here in detail. Chapter 3 describes the mathematics of cryptography. Discrete logarithm problem (DLP) and cryptographic hash functions are discussed here. Chapter 4 describes about the proposed signature scheme. In chapter 5, a new e-voting protocol based on the proxy blind signature is discussed.

Chapter 2

Literature Review

2.1 Related Work

On the basis of Schnorr blind signature, Tan et al. [7] proposed the concept of proxy blind signature, having the advantages of both the proxy signature and the blind signature. This scheme was based on the discrete logarithm problem. Later, Lal et al. [8] pointed out that Tan et al.'s proxy blind signature scheme suffer from a kind of forgery attack due to the signature receiver. Compared with Tan et al.'s scheme, Lal et al. further proposed a more efficient and secure proxy blind signature scheme to overcome the pointed out drawback in Tan et al.'s scheme. Sun et al. [9] show that Tan et al.'s scheme does not satisfy the unforgeability and unlinkability properties. In addition, they also point out that Lal et al.'s scheme does not possess the unlinkability property. But they did not give an improved scheme to overcome the insecurity. Wang et al. [10] demonstrated that Tan et al.'s scheme was insecure and proposed two effective attacks. Later, wang et al. [11] showed three security threats in Tan et al.'s scheme and proposed the remedy for that. In 2004, Xue et al. [12] showed there exists one weakness in Tan et al.'s scheme and Lal et al.'s scheme since the proxy signer can get the link between the blind message and the signature or plaintext with great probability. Xue et al. introduced concept of strong unlinkability and they also proposed a proxy blind signature scheme. Compared with Tan et al.'s scheme and Lal et

al.'s scheme, their scheme is more efficient. However, Li et al. [13] showed xue et al.'s scheme cannot satisfy unforgeability and strong unlinkability properties. Later, Yang et al. [34] proposed an efficient proxy blind signature scheme based on discrete logarithmic problem and proved that their scheme is more secure and efficient than other existing schemes.

2.2 Review of Tan et al. Shceme

For the convenience of describing the scheme, the following parameters are defined as follows:

Alice : Original Signer

Bob : Proxy Signer

R : Receiver

p : a large prime number

q : a prime factor of p-1

g : an element of Z_p^*

$x_A, x_B, x_R \in Z_q^*$: the original signer Alice's secret key,the proxy signer Bob's secret key, the receiver R's secret key.

$y_A \equiv g^{x_A} (\text{mod p})$: Alice's public key

$y_B \equiv g^{x_B} (\text{mod p})$: Bob's public key

$y_R \equiv g^{x_R} (\text{mod p})$: R's public key

H(.) : a public cryptographically strong hash function

|| : which denotes the concatenation of strings

2.2.1 Proxy Phase

Commission Generation

Alice randomly chooses $\bar{k} \in Z_q^*$ and computes

$$\bar{r} = g^{\bar{k}} \ (mod\ p) \tag{2.1}$$

$$\bar{s} = x_A \bar{r} + \bar{k} \ (mod\ p) \tag{2.2}$$

Proxy Delivery

Alice gives the pair $(\bar{r}, \bar{s})$ to the proxy signer, Bob via secure channel.

Proxy Verification

Bob checks,

$$g^{\bar{s}} = \bar{r} y_A^{\bar{r}} \ (mod\ p) \tag{2.3}$$

which is often called as delegation function. If it is correct, Bob accepts and computes

$$s' = \bar{s} + x_B \ (mod\ p) \tag{2.4}$$

2.2.2 Signing Phase

Bob chooses a random number, $k \in Z_q^*$, and computes

$$t = g^k \ (mod\ p) \tag{2.5}$$

and sends $(\bar{r}, t)$ to the receiver R. R chooses two random numbers $a, b \in Z_q^*$, and computes

$$r = tg^b y_B^{-a-b}(\bar{r}y_A^{\bar{r}})^{-a} \ (mod\ p) \tag{2.6}$$

$$e = H(r||m) \quad (mod\ q) \tag{2.7}$$

$$u = (\bar{r}y_A^{\bar{r}})^{-e+b} y_A^{-e} \ (mod\ p) \tag{2.8}$$

$$e^* = e - a - b \ (mod\ q) \tag{2.9}$$

if r=0, then R needs to select a new tuple (a,b) otherwise, R delivers e^* to the proxy B.

After receiving e^*, Bob computes

$$s'' = e^* s' + k \ (mod\ q) \tag{2.10}$$

then Bob sends s'' to R.

2.2.3 extraction phase

While receiving s'', R computes

$$s = b + s'' \ (mod\ q) \tag{2.11}$$

Then, the proxy blind signature is the tuple(m,u,s,e).

2.2.4 Verification

The recipient of a proxy blind signature can verify its validity by checking that

$$e \stackrel{?}{=} H(g^s y_B^{-e} y_A^e u||m) \ (mod\ q) \tag{2.12}$$

2.3 Security Analysis of Tan et al. Proxy Blind Signature Scheme

In this section, the security shortcomings of Tan et al. scheme are analyzed.

2.3.1 The original signer's universal forgery attack

A malicious original signer can forge a proxy blind signature [11,14] by computing

$$g^s y_B^{-e} y_A^e u(mod\ p) = t g^b (y_A^{\bar{r}'} \bar{r}')^{-a} y_B^{-a-b}\ (mod\ p) \tag{2.13}$$

By computing using (2.5) to (2.11), he has

$$g^s y_B^{-e} y_A^e u(mod\ p) = g^{b+s''} y_B^{-e} y_A^e (y_A^{\bar{r}'} \bar{r}')^{-e+b} y_A^{-e}\ (mod\ p) \tag{2.14}$$

$$= t g^b g^{(e-a-b)s'_A} y_B^{-e} (y_A^{\bar{r}'} \bar{r}')^{-e+b}\ (mod\ p) \tag{2.15}$$

So, (2.13) follows from the equation

$$g^{(e-a-b)s'_A} y_B^{-e} (y_A^{\bar{r}'} \bar{r}')^{-e+b} (mod\ p) = (y_A^{\bar{r}'} \bar{r}')^{-a} y_B^{-a-b}\ (mod\ p) \tag{2.16}$$

By simplifying further (2.16), he has

$$g^{s'_A} (mod\ p) = (y_A^{\bar{r}'} \bar{r}') y_B\ (mod\ p) \tag{2.17}$$

The malicious original signer can easily create suitable s'_A and $\bar{r}'$,
(For example, he chooses randomly $v \in Z_q^*$, then creates $\bar{r}' = y_B^{-1} g^v\ (mod\ p)$ and $s'_A = x_A \bar{r}'\ (mod\ q)$) by using (2.17), then he can forge a proxy blind signature using s'_A .

2.3.2 The receiver's universal forgery attack

After receiving the valid signature (m,u,s,e) on message m, suppose a receiver R wants to forge a valid proxy blind signature (m', s', s, e_f) on message m' he chooses arbitrarily [8,11,15], he perform as follows.
computes $e_f = H(r||m')\ (mod\ q)$
computes u', R can get u' by computing the following equation

$$g^s y_B^{-e_f} y_A^{e_f} u'(mod\ p) = t g^b y_B^{-a-b} (y_A^{\bar{r}})^{-a}\ (mod\ p) \tag{2.18}$$

By computing using (2.5) to (2.11), he has

$$g^s y_B^{-e_f} y_A^{e_f} u'(mod\ p) = g^{e^* s' + k + b} y_B^{-e_f} y_A^{e_f} u'\ (mod\ p) \tag{2.19}$$

$$= t g^b (y_A^{\bar{r}} \bar{r})^{e-a-b} y_B^{e-a-b} y_B^{-e_f} y_A^{e_f} u'\ (mod\ p) \tag{2.20}$$

So, (2.18) follows from the equation

$$tg^b(y_A^{\bar{r}}\bar{r})^{e-a-b}y_B^{e-a-b}y_B^{-e_f}y_A^{e_f}u' \ (mod\ p) = tg^b y_B^{-a-b}(y_A^{\bar{r}}\bar{r})^{-a} \ (mod\ p) \tag{2.21}$$

By simplifying further (2.21), he has

$$u' = (y_A^{\bar{r}}\bar{r})^{-e+b}y_B^{e_f-e}y_A^{-e_f} \ (mod\ q) \tag{2.22}$$

Therefore, R can forge a valid signature (m', u', s, e_f) on message m' he chooses arbitrarily.

2.3.3 Linkability Attack

Suppose the proxy signer Bob holds the signature sig($\overline{m}$) on blind message $\overline{m}$ and related parameters, he can figure out the random numbers a and b by (2.9) and (2.11) after knowing a proxy signature tuple (m,u,s,e) [11, 15]. Here a and b are random numbers secretly chosen by the user, which should not be known to others in a blind signature scheme due to the blindness requirement.

2.4 Review of Lal et al. Scheme

The notations are same as the previous scheme (Tan et al. scheme). The proposed scheme is divided into three phases.

2.4.1 Proxy Phase

Proxy Generation

The original signer, Alice randomly chooses $k \in Z_q^*$, $k \neq 1$ and computes

$$r = g^k \ (mod\ p) \tag{2.23}$$

$$\bar{s} = x_A + kr \ (mod\ q) \tag{2.24}$$

$$y_p = g^s y_B \ (mod\ p) \tag{2.25}$$

Proxy Delivery

The original signer, Alice sends $(\bar{s}, r)$ to the proxy signer, Bob in a secure way and makes y_p public.

Proxy Verification

After receiving $(\bar{s}, r)$ the proxy signer, Bob checks the validity of the following congruence

$$y_p = g^{\bar{s}} = y_A r^r \ (mod\ p) \tag{2.26}$$

If $(\bar{s}, r)$ satisfies this congruence, he accepts it and computes

$$s = \bar{s} + x_B \ (mod\ q) \tag{2.27}$$

as his/her proxy private key.

2.4.2 Signing Phase

Bob chooses a random number $k \in Z_q^*$, $k \neq 1$, and computes

$$t = g^k \ (mod\ p) \tag{2.28}$$

and sends it to the receiver, R.
R chooses randomly $\alpha, \beta \in Z_q^*$ and computes

$$r' = tg^{-\alpha} y_p^{-\beta} \ (mod\ p) \tag{2.29}$$

If r'=0, he chooses another set of α and β; otherwise computes

$$e' = H(r' \oplus m) \ (mod\ q) \tag{2.30}$$

$$e = e' + \beta \ (mod\ p) \tag{2.31}$$

and R sends e to Bob.
After receiving e, Bob computes

$$s' = k - se \ (mod\ q) \tag{2.32}$$

and sends it to receiver, R.

Now R computes,

$$s_p = s' - \alpha \ (mod\ q) \tag{2.33}$$

The tuple (m,s_p, e') is the proxy blind signature.

2.4.3 Verification Phase

The verifier or recipient of the proxy blind signature computes

$$e'' = H(g^{s_p} y_p^{e'} \ (mod\ p) \oplus m) \ (mod\ q) \tag{2.34}$$

Here, $e'' = e'$, if and only if the tuple (m,s_p, e') is a valid proxy signature.

2.5 Security Analysis of Lal et al. Scheme

In this section the attacks on Lal et al. Sheme are analyzed.

2.5.1 Linkability Attack

For the proxy signer, in order to identify the relationship between the revealed message and the blind information, the proxy signer records all messages he owned, such as t(s), e(s), and $s'(s)$. After a signature (m, s, e') is revealed, the proxy signer computes

$$a' = s' - s \tag{2.35}$$

$$b' = e - e' \tag{2.36}$$

$$r' = g^s y_{pr}^{e'} \ (mod\ p) \tag{2.37}$$

for some $s' \in s'(s)$ and e $\in$ e(s).

Finally, the proxy signer checks the equation

$$r' = t g^{-a'} y_p^{-b'} (mod\ p), \quad for\ some\ t \in t(s) \tag{2.38}$$

If he finds a corresponding t such that $r' = tg^{-a'}y_p^{-b'}(mod\ p)$, therefore, the proxy signer knows that (t,e,s'), is the related blind information corresponding to the revealed message m. So, Lal et al's proxy blind signature does not possess the unlinkability property [12].

2.5.2 Attack on the publishing of the proxy public key

In order to verify a proxy signature, the proxy public key is obtained by computing, while not retrieving from original signers publishing. The computed proxy public key has the meaning of confirming the relationship between a original signer and a proxy signer. In Lal and Awasthis scheme, such a publishing enables an adversary who obtained the proxy public key to republish it again. Finally, the adversary claims that he is the original signer. Therefore, the publishing of proxy public key suffers from the security flaw that the original signer is unable to be authenticated exactly [9].

2.6 Review of Xue et al. Scheme

In 2004, Xue et al. [12]. proposed a new proxy protected proxy blind signature scheme with warrant. In this scheme, the CA (Certificate Authority) is needed. Its task is to manage the public directory in the system and certify users' public keys. The scheme is divided into the following subsections:

2.6.1 Proxy Phase

Proxy Generation

Original signer, Alice selects $\bar{k} \in Z_q^*$ at random and computes

$$\bar{r} = g^{\bar{k}} \tag{2.39}$$

$$\bar{s} = \bar{k} + x_A H(m_w, \bar{r})\ (mod\ q) \tag{2.40}$$

Proxy Delivery

Alice sends the pair $(m_w, \bar{r}, \bar{s})$ to the proxy signer B.

2.6.2 Proxy Verification

Bob checks whether the following equation holds or not.

$$g^{\bar{s}} = \bar{r} y_A^{H(m_w, \bar{r})} \ (mod\ p) \tag{2.41}$$

If it holds, Bob continues to compute

$$s' = \bar{s} + x_B y_B \ (mod\ q) \tag{2.42}$$

$$y_p = g^{s'} \tag{2.43}$$

$$= g^{\bar{s}} y_B^{y_B} \ (mod\ p) \tag{2.44}$$

$$= \bar{r} y_A^{H(m_w, \bar{r})} y_B^{y_B} \ (mod\ p) \tag{2.45}$$

as his/her secret and public proxy signature key, respectively.

2.6.3 Signing Phase

Bob selects $k \in Z_q^*$ at random, and computes

$$t = g^k \ (mod\ p) \tag{2.46}$$

and then sends t to the receiver R.
R chooses two random integers $a, b \in Z_q^*$, and calculates

$$r = t g^{-a} y_p^{-b} \ (mod\ p) \tag{2.47}$$

If r=0, R rechooses a and b. Once r, a, and b are determined, the receiver R computes

$$e' = H(r||m) \ (mod\ q) \tag{2.48}$$

$$e = e' + b \ (mod\ q) \tag{2.49}$$

Then R delivers e to the proxy signer B.

After receiving e, B calculates

$$s'' = k - s'e \ (mod\ q) \tag{2.50}$$

Then, B sends $(m_w, \bar{r}, s'')$ to receiver R.

2.6.4 Extraction Phase

While receiving s'', R computes

$$s = s'' - a \ (mod\ q) \tag{2.51}$$

$$S = g^s \ (mod\ p) \tag{2.52}$$

Then, the proxy blind signature is the tuple $(m, m_w, \bar{r}, S, e')$.

2.6.5 Verification Phase

From m_w, the recipient of a proxy blind signature can get the public keys of the original signer and proxy signer, the delegation time, etc. Then he/she, can get the public keys of the original signer and the proxy signer from CA.

The recipient of a proxy blind signature can confirm its validity by checking that

$$e' \stackrel{?}{=} H(S(\bar{r}y_A^{H(m_w,\bar{r})}y_B^{y_B})e'(mod\ p)||m) \ (mod\ q) \tag{2.53}$$

2.7 Security Analysis of Xue et al. Scheme

In 2005, Li et al. [13] proved that Xue et al. scheme failed to satisfy the unforgeability and strong unlinkability property.

2.8 Review of Yang et al. Scheme

In 2008, Yang et al. [34] proposed a new proxy blind signature scheme, which satisfied all the security requirements of both the blind signature scheme and the

proxy signature scheme.

Here it is assumed that the proxy signer, Bob will blind sign a message m on behalf of the original signer Alice, the receiver is R.

The proposed scheme is divided into into five phases:

1. system setup
2. proxy designate
3. blind signing
4. signature extraction
5. signature verification

2.8.1 System Setup

The parameters are defined as follows:

Alice : Original signer

Bob : Proxy signer

R : Receiver

p, q : two large prime numbers, such that q | p-1

g : an element of Z_q^*, its order is q.

m_w : the designated proxy warrant which contains the identities information of the original signer and the proxy signer, message type to be signed by the proxy signer, the delegation limits of authority, valid periods of delegation, etc.

$x_A , x_B \in Z_q^*$: the original signer Alice's secret key, the proxy signer Bob's secret key.

$y_A = g^{x_A} \pmod p$: Alice's public key.

$y_B = g^{x_B} \pmod p$: Bob's public key.

H(.), h(.) : public cryptographically strong hash functions.

|| : the concatenation of strings.

2.8.2 Proxy Designation

Alice selects $\bar{k} \in Z_q^*$, and computes

$$K = g^{\bar{k}} \ (mod\ p) \tag{2.54}$$

$$\bar{s} = x_A + \bar{k}.H(m_w||k) \ (mod\ q) \tag{2.55}$$

Alice sends $(K, \bar{s})$ along with the warrant m_w to the proxy signer Bob via a secure channel.

Bob checks the equation

$$g^{\bar{s}} = y_A K^{H(m_w||K)} \ (mod\ p) \tag{2.56}$$

If it is correct, Bob accepts the proxy task and computes

$$s' = \bar{s} + x_B \tag{2.57}$$

as his proxy blind signature secret key.

2.8.3 Blind Signing

Bob selects $k \in Z_q^*$, and computes

$$t = g^k \ (mod\ p) \tag{2.58}$$

and then sends (K,t) to the receiver R. R randomly selects two numbers $a, b \in Z_q^*$, and computes

$$r = t^a (y_A y_B K^{H(m_w||K)})^{ab} \ (mod\ p) \tag{2.59}$$

$$e = h(m||r) \ (mod\ q) \tag{2.60}$$

$$e' = a^{-1} e + b \ (mod\ q) \tag{2.61}$$

If r=0, R has to select a new tuple (a,b). R sends e' to Bob.

After receiving e', Bob computes

$$s'' = e's' + k \tag{2.62}$$

and sends the signed messages s'' to R.

2.8.4 Signature Extraction

After receiving s'', receiver R computes

$$s = g^{s"a} \ (mod\ p) \tag{2.63}$$

Finally, the proxy blind signature scheme is the tuple (m, m_w, s, e, K).

2.8.5 Signature Verification

The verifier can verify the validity of the proxy blind signature by checking that

$$e = h(m||s\ (y_A y_B K^{H(m_w||K)})^{-e})\ (mod\ q) \tag{2.64}$$

2.9 Security Analysis of Yang et al. Scheme

The Yang et al. proxy blind signature scheme is not secure against forgeability attack. An attacker can create a valid proxy blind signature instead of the designated proxy signer.

2.9.1 Forgeability Attack

An attacker, E can produce a proxy signature instead of Bob, who is delegated by the original signer Alice. The attacker go through the following steps to produce a valid proxy blind signature.

- Step 1. E chooses a forged message m' to be signed by him/her.
- Step 2. E randomly selects two integers k, $k' \in Z_p^*$.
- Step 3. E then computes the followings

$$K' = g^{k'}\ (mod\ p) \tag{2.65}$$

$$t' = g^{k}\ (mod\ p) \tag{2.66}$$

$$e' = h(m'||t')\ (mod\ q) \tag{2.67}$$

$$s' = t'.(y_A y_B K'^{H(m_w||K')})^{e'}\ (mod\ p) \tag{2.68}$$

- Step 4. The forged blind signature on message m' is (m', m_w, s', e', K').

The generated forged blind signature is valid and it can be verified by the verifier Bob as follows:

$h(m'||s'(y_A y_B K'^{H(m_w||K')})^{-e'})\ (mod\ q)$

$=h(m'||t'(y_A y_B K'^{H(m_w||K')})^{e'})(y_A y_B K'^{H(m_w||K')})^{-e'}\ (mod\ q)$

$=h(m'||t')\ (mod\ q)$

$=e'$

It is proved that an attacker can produce the proxy blind signature on the forged message m' [14, 19]. So, Yang et al. scheme is not secure against forgeability attack.

2.10 Observation

So finally it has been observed that, the proxy blind signature schemes stated above failed to satisfy all the security properties of proxy blind signature scheme that are discussed in the Introduction chapter. So the objective is to propose a new proxy blind signature scheme with minimum computational cost and it should satisfy all the security requirements of a proxy blind signature.

2.11 Problem Definition

During the verification of a proxy blind signature scheme the verifier cannot know whether signing (done by proxy signer) is within the delegation period or not. Proxy signer can make fool to the verifier by signing the message or document after the delegation period is over as there is no such provision to record the time stamp during the proxy signing phase. Original signer cannot revoke the delegation whenever necessary, so that a proxy signer may misuse the delegating power for signing. Hence, it is necessary to provide a time stamp during the signing phase of the proxy blind signature and to allow the original signer to revoke delegating power whenever necessary.

2.12 Objective

The objectives are:

- To design a new proxy blind signature scheme with proxy revocation.
- To provide a time stamp in the signing phase so that a verifier can know the signing was done within the delegation period.
- To compare the proposed scheme with the existing scheme based on efficiency and computational time.

Chapter 3

Mathematical Background

3.1 Discrete Logarithm Problem

The Discrete Logarithm Problem is a critical problem in number theory, and is similar in many ways to the integer factorization problem. Discrete logarithms were used mainly in computations of finite fields and elliptic curves. Discrete logarithm problem has significant importance in the field of cryptography as the complexity lies in solving the discrete logarithm problem. If it were possible to compute discrete logs efficiently, it would be possible to break numerous thought-to-be unbreakable cryptographic schemes. To define a discrete logarithm one picks an element g in the field and then one picks a secret random integer x and one computes $h = g^x$ in the field. The discrete logarithm problem is given g and h, find x.

Discrete Logarithm Problem is a good source of a one-way function. A one-way function as a function $f : X \to Y$ for which given $x \in X$ it is easy to compute f(x); however, given $y \in Y$; it is difficult to compute a value $x \in X$ such that f(x)=y, at least for most values of y. In other words, the function f is not invertible, without further information, and it is for this reason that such function is otherwise known as a trapdoor function.

3.2 Hash Function

A cryptographic hash function takes a message of arbitrary length and creates a message digest of fixed length such that any (accidental or intentional) change to the message will change the hash value (message digest) with very high probability. The ideal cryptographic hash function has four main properties:

- It is easy to compute the hash value for any given message.
- It is infeasible to generate a message that has a given hash.
- It is infeasible to modify a message without changing the hash.
- It is infeasible to find two different messages with the same hash.

In various standards and applications, the two most commonly used hash functions are MD5 and SHA-1.
The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a message digest that is 128 bits long. The digest is sometimes also called the "hash" or "fingerprint" of the input. MD5 is used in many situations where a potentially long message needs to be processed and/or compared quickly. The most common application is the creation and verification of digital signatures. SHA-1 is a widely used cryptographic hash function developed by the NSA. It's result is usually expressed as a 160 bit hex number. SHA-1 is widely considered the successor to MD5.

3.3 Group

A group is a finite or infinite set of elements together with a binary operation (called the group operation) that together satisfy the four fundamental properties of closure, associativity, the identity property, and the inverse property. The operation with respect to which a group is defined is often called the *group operation*, and a set is said to be a group under this operation. Elements a,

b, c , ... of a set G with binary operation between a and b denoted a*b form a group G if the following four properties are satisfied,

1. Closure

   If a and b are two elements in G, then a*b is also in G. It's called closed because from inside the group, we can't get outside of it.

2. Associativity

   For all a, b, and c in G, a * (b * c) = (a * b) * c.

   It means that the order in which we do operations doesn't matter.

3. Identity

   There exists an identity element e in the set G, such that a * e = a

   and e * a = a, for all elements a in G. There is only one identity element for every group.

4. Inverse

   If we have an element of the group, there is another element of the group such that when we use the operator on both of them, we get e, the identity.

   For all a in G, there exists b in G, such that a * b = e and b * a = e.

Order

The order $|G|$ of a finite group G is the number of elements of G.

The order of an element g in a group is the least positive integer k such that g^k is the identity.

Chapter 4

Proposed Proxy Blind Signature Scheme

In this chapter a new and improved proxy blind signature scheme with proxy revocation is proposed which satisfies all the security requirements of the proxy blind signature scheme. The proposed scheme also records the time stamp of the signing phase so that a verifier can get sure that the signing is done within the delegation period.

4.1 Proposed Scheme

The proposed scheme is divided into following phases:

1. System parameter initialization
2. Proxy delegation
3. Blind signing
4. Signature extraction
5. Signature verification

4.1.1 System Parameter Initialization

The parameters used in the proposed scheme are:

Alice : Original Signer

Bob : Proxy Signer

R : Signature Requester

AS : Authentication Server as trusted third party

p, q : two large prime numbers such that, q$|p-1$

g : an element of order q in Z_p^*

$x_A, x_B, x_R \in Z_q^*$: the original signer Alice's secret key, the proxy signer Bob's secret key, and R's secret key respectively.

$y_A = g^{x_A}(mod\ p)$: Original signer Alice's public key

$y_B = g^{x_B}(mod\ p)$: Proxy signer Bob's public key

$y_R = g^{x_R}(mod\ p)$: Receiver R's public key

H(.) : a cryptographically secure one way hash function

|| : which denote the concatenation of two strings

m_w : message warrant

m : message

4.1.2 Proxy Delegation

The original signer Alice randomly picks out $\bar{k} \in Z_q^*$ and computes,

$$r = g^{\bar{k}}\ (mod\ p) \tag{4.1}$$

$$s = x_A + \bar{k}.H(m_w||r)\ (mod\ q) \tag{4.2}$$

Alice sends (r, s) along with the message warrant m_w to the proxy signer Bob and AS, via a secure channel.

The proxy signer Bob, then verifies the equation

$$g^s = y_A r^{H(m_w||r)}(mod\ p) \tag{4.3}$$

If it is correct, Bob accepts and computes,

$$s_{pr} = s + x_B y_A \tag{4.4}$$

as his/her proxy blind signature secret key.

4.1.3 Blind Signing

Proxy signer, Bob randomly selects an integer $k \in Z_q^*$, and computes

$$t = g^{k+x_B}(mod\ p) \tag{4.5}$$

Bob, then sends (r, t, m_w) to the receiver R.

R checks Alice's and Bob's identities and the delegation lifetime of the warrant m_w.

If the above checking is successful,

R selects two random numbers $u, v \in Z_q^*$ and computes

$$r' = tg^{u+x_R}y_{pr}^{v}\ (mod\ p) \tag{4.6}$$

where x_R is the private key of R and $y_{pr} = g^{s_{pr}}\ (mod\ p)$

$$e = H(r'||m)\ (mod\ q) \tag{4.7}$$

$$e^* = v - e\ (mod\ q) \tag{4.8}$$

If r'=0, then R needs to select a new tuple (u, v) otherwise, R sends e^* to Bob and AS.

For signing blinded message, Bob must request a time stamp for the message. Bob transmits his identity and (s, m_w, t) to AS. AS checks whether the received s from Bob and the received s from Alice is identical. If these two are same then AS checks

$$g^s = y_A r^{H(m_w||r)}(mod\ p) \tag{4.9}$$

If it satisfies, AS goes through the following steps:

1. It is still in the valid proxy delegation specified in m_w.
2. r is not in the revocation list. If r is in the revocation list, then it means that the delegation is revoked.

After that, AS chooses a random number $k_s \in Z_q^*$ and computes

$$r_s = g^{k_s} \ (mod\ p) \tag{4.10}$$

$$T = H(r_s \ || \ time\ stamp \ || \ e^*) \ (mod\ p) \tag{4.11}$$

AS sends T to the proxy signer Bob and receiver R.
After receiving T, the proxy signer Bob computes,

$$s' = k + e^* s_{pr} + T \tag{4.12}$$

as the signed message and sends it to the receiver R.

4.1.4 Signature Extraction

After receiving s' from Bob, the receiver R computes,

$$s^* = g^{u+s'-T} (mod\ p) \tag{4.13}$$

Thus, the proxy blind signature on message m is the tuple (m, m_w, s^*, e).

4.1.5 Signature Verification

Verifier can verify the proxy blind signature by checking whether

$$e \stackrel{?}{=} H(s^* y_B y_R y_{pr}^e || m) \ (mod\ q) \tag{4.14}$$

4.1.6 Revocation Phase

If original signer Alice wants to revoke the delegation before the specified delegation period , then Alice ask AS to put r in the revocation list. During the computation of T, AS checks the validity of delegation period specified in the proxy warrant m_w and the revocation list. If it is within the valid delegation period and r is not found in the revocation list, AS computes T, sends it to Bob and R for the message. If r is in the revocation list then AS does not compute T. Hence, the proxy signer, Bob cannot sign. r in the revocation list can be removed after the delegation period is over. Therefore, the size of the revocation list will not be unlimited.

4.2 Security Analysis of the Proposed Scheme

1. If the original signer have an intention to forge a proxy blind signature with forgery attack for the message m', he/she has to create a secret key s'_{pr} and calculate

$$y'_A = g^{s'_{pr}} \ (mod\ p) \tag{4.15}$$

Consequently, the original signer must compute

$$s^* y_B y_R y_{pr}^e \ (mod\ p) = t g^{u+x_R} y_{pr}^v \ (mod\ p) \tag{4.16}$$

By using the equation (4.5) to (4.10), the original signer has

$$g^{u+s'-T+x_B+x_R} y_{pr}^e (mod\ p) = t g^{u+x_R} y_{pr}^v \ (mod\ p) \tag{4.17}$$

$$\Rightarrow g^{v-e} s'_{pr} (mod\ p) = y_{pr}^{v-e} \ (mod\ p) \tag{4.18}$$

To find the value of s'_{pr} original signer must find a solution to the above equation (4.15) which is a discrete logarithm problem. Thus, the original signer fails to forge a signature.

2. The receiver can not forge the signature after receiving (m, m_w, s^*, e) on message m. When a receiver tries to forge a signature (m', s^*, e') for message m', he/she must verify that the equation given below is correct.

$$s^* y_B y_R y_{pr}^e \ (mod\ p) = t g^{u+x_R} y_{pr}^v \ (mod\ p) \tag{4.19}$$

By using the the equations(4.5) to (4.10) he has

$$s^* y_B y_R y_{pr}^e \ (mod\ p) = g^{u+s'-T} g^{x_B} g^{x_R} y_{pr}^{e'} \ (mod\ p) \tag{4.20}$$

$$= g^{u+s'-T+x_B+x_R} g^{s_{pr}e'} \ (mod\ p) \tag{4.21}$$

$$= t g^{u+(v-e)s_{pr}+x_R} g^{s_{pr}e'} \ (mod\ p) \tag{4.22}$$

$$= t g^{u+x_R} y_{pr}^v \tag{4.23}$$

From the above we can get,

$$g^{(v-e)s_{pr}} g^{s_{pr}e'} \ (mod\ p) = g^{s_{pr}v} \ (mod\ p) \tag{4.24}$$

This cannot hold true, as $e \neq e'$. Therefore the receiver fails to forge a valid proxy blind signature on message m'.

3. The proxy linkability holds if there is a conjunction between (t, e^*, s') and (m, m_w, s^*, e). t is only in equation (4.6) and relate to e through equation (4.7). Proxy signer cannot find out the value of t as it is masked by two random numbers u and v. Hence, the proposed scheme satisfies the unlinkability property.

4. As the proxy blind signature (m, m_w, s^*, e) on the message m, contains m_w(message warrant) anyone can easily differentiate between the proxy blind signature and normal signature. Hence, it satisfies the distinguishability property.

5. From the warrant m_w, anyone can mark original signer and proxy signer. On the other hand, as the verification equation contains the public key of the proxy signer and original signer, one can identify them. As a result, anyone can determine the identity of the corresponding proxy signer from a proxy signature. Hence, it satisfies identifiability property.

6. The original signer cannot get the proxy signer's secret key, and similarly the proxy signer cannot get the original signer's secret key. So, one cannot sign on behalf of other. Hence, it satisfies the non repudiation property.

7. Due to the inclusion of the original signer and proxy signer identities information, message type to be signed by the proxy signer, delegation period, etc. in the warrant itself the proposed scheme is capable of preventing proxy key pair misuse.

8. verification

   The proposed scheme satisfies the property of verifiability.

   $H(s^* y_B y_R y_{pr}^e || m) \ (mod\ q)$
   $= H(s^* g^{x_B} g^{x_R} y_{pr}^e || m) \ (mod\ q)$
   $= H(g^{s'+u-T+x_B+x_R} y_{pr}^e \ || m) \ (mod\ q)$

$= H(g^{k+x_B+u+x_R+e^* s_{pr}} y_{pr}^e \,||m)\ (mod\ q)$

$= H(g^{k+x_B+u+x_R+(v-e)s_{pr}} y_{pr}^e \,||m)\ (mod\ q)$

$= H(g^{k+x_B+u+x_R+s_{pr}v-s_{pr}e} y_{pr}^e \,||\, m)\ (mod\ q)$

$= H(g^{k+x_B+u+x_R+s_{pr}v} y_{pr}^e y_{pr}^{-e} \,||\, m)\ (mod\ q)$

$= H(g^{k+x_B+u+x_R+s_{pr}v} \,||\, m)\ (mod\ q)$

$= H(g^{k+x_B} g^{u+x_R} y_{pr}^v \,||\, m)\ (mod\ q)$

$= H(tg^{u+x_R} y_{pr}^v || m)\ (mod\ q)$

$= H(r'||m)\ (mod\ q)$

=e

4.3 Efficiency Analysis of the Proposed Scheme

Let M and E denote computational load for multiplication and exponentiation respectively. The computational load for addition is ignored due to its high performance. The table given below gives the detail comparison of computational loads of the proposed scheme with other existing schemes.

Schemes	Proxy Generation	Blind Signing	Verification	Total
Tan et al.	4E+3M	7E+6M	3E+3M	14E+12M
Lal et al.	4E+3M	3E+3M	2E+M	9E+7M
Xue et al.	3E+3M	4E+3M	3E+3M	10E+9M
Yang et al.	3E+2M	5E+4M	2E+3M	10E+9M
Proposed Scheme	3E+3M	6E+3M	E+3M	10E+9M

Chapter 5

A Secure E-voting Protocol Based on Proxy Blind Signature

5.1 Introduction

Voting is a way for a voter to make a decision or express an opinion or to choose a candidate. E-voting (Electronic voting) refers to both the electronics means of casting a vote and the electronic means of counting and publishing that votes. E-voting system has some specific advantages as compared to the traditional voting system. Many people are not going to vote as because voting booth is far away from their work place. The only solution to it is e-voting scheme. E-voting has become increasingly popular in our technology driven world. It increases the security of the ballot, speed up the processing of results and make voting easier. E-voting also has the ability to reduce fraud, by eliminating the opportunity for ballot tampering. Due to mobility and convenience, the most important properties of e-voting, it is becoming more popular [26, 32, 33].
In general, two main types of e-voting can be identified:

1. E-voting which is physically supervised by representatives of government or independent electoral authorities. (e.g. electronic voting machines located at polling stations)

2. Remote e-voting where voting is performed within the voters sole influence, and is not physically supervised by representatives of government authorities. (e.g. voting from one's personal computer, mobile phone)

E-voting is an election system that allows a voter to record his or her secure and secret ballot electronically. E-voting can reduce election costs and increase participation of voters by making the voting process more convenient.

5.2 Security Properties of E-voting

1. Completeness

   In traditional voting scheme the voters identity is checked by seeing the voter in person. But in e-voting, the voter has to pass a serial of authentication procedures after that he/she is permitted to cast his/her vote. Completeness property says that only authorized voters are eligible to vote.

2. Accuracy

   A vote cannot be altered, cannot be eliminated from counting, invalid vote should not be counted.

3. Uniqueness

   A voter can vote exactly once, more than once is avoided.

4. Privacy

   The definition of privacy states that no one can determine how an individual voter gave its vote. Voters also cannot prove how they have voted, otherwise they may sell their vote.

5. Reliability

   During major failures (e.g. internet failure) the system should be robust and no loss of vote should happen.

6. Verifiability

   This property states that each voter can verify that their vote is correctly counted.

7. Mobility

   Mobility is one of the basic properties of important of e-voting. It states that voters are not physically restricted to cast their vote.

8. Fairness

   The properties of fairness states that, no one can get the voting result before its publication phase. Fairness is always regarded as an essential for preventing vote-buying.

9. Anonymity

   The definition of anonymity in e-voting states that no one can link the voted ballot to the voter who has cast that vote.

10. Convenience

    It states that the voters cast their votes quickly and with minimal skills. The system should be user friendly.

11. Robustness

    The robustness property defines that no attacker or dishonest voter can disturb or interrupt the voting process.

12. Efficiency

    The property of efficiency states that the voting scheme should produce a specific result effectively within a minimum amount of time and voters are not required to wait for other voters to complete the process.

5.3 Proposed Scheme

In the proposed scheme there are only four participants involved as follows:

1. Registration Authority (RA):

   RA is a trusted party where all the eligible voter have to register in advance.

2. Administrator (A):

   Administrator monitors the whole process of the voting scheme.

3. Vote Counter (VC):

   VC has the responsibility to count the valid votes and publish the result.

4. Voter (V_i):

   Voter is someone who is eligible to give the vote.

5.3.1 Structure of the Proposed Scheme

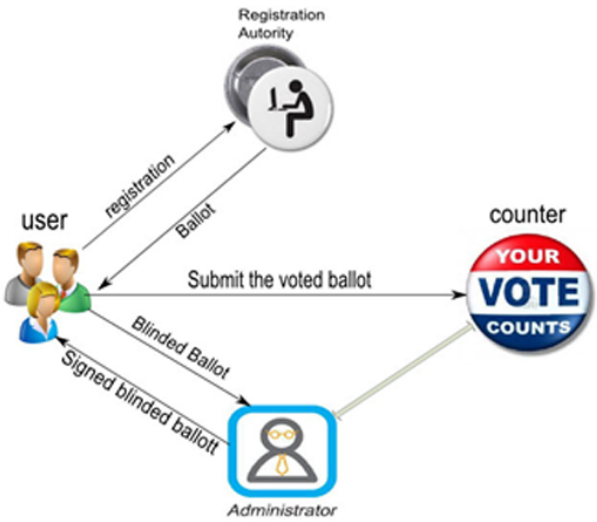

Figure 5.1: Structure of the proposed scheme

Every participants i.e. every voter, registration authority (RA), administrator (A), and vote counter (VC) generate their public key and private key individually in advance. Everyone get the public key of others from the certification authority

(CA), by a secure authorized channel. The proposed scheme is divided into mainly 3 phases:

1. Registration

   Voter V_i send an encrypted message to registration authority (RA) requesting for registration. The message contains the ID of the voter V_i. After receiving the message, registration authority (RA) verifies the authenticity of the sender V_i and check the voting right of voter V_i. RA also checks that whether V_i has applied previously or not. With proper verification the RA sends ballot papers to voter V_i.

2. Voting

   Voter V_i fills the ballot, makes blind using blind signature technique and sends to the administrator (A) to get his/her signature on the blinded ballot. Administrator (A) signs the hidden ballot and returns back to the voter.

3. Counting

   Voter V_i sends the signed ballot, hash value of unique number from RA to the vote counter anonymously. After the voting deadline is over, vote counter (VC) publishes the result.

5.3.2 Proposed Scheme in Detail

Registration

At the beginning, the voter V_i sends an encrypted message to RA by using his secret key. The message contains ID of voter, a random number (rn), ID of administrator, time stamp. After getting the message, RA first checks the authenticity of the message and then checks whether the voter V_i is eligible to vote then RA checks whether V_i has applied for registration or it is first time. If voter V_i is authenticated properly, a unique vote number NV_i is generated by RA. Then RA sends the encrypted message to voter V_i, $E_{RAs}(IDV_i \mid\mid NV_i \mid\mid rn-1 \mid\mid time\ stamp)$

Voting

The following parameters are used in this phase:

p, q : two large prime numbers such that, q|$p-1$

g : an element in Z_p^* whose order is q

$x_C, x_A \in Z_q^*$: the Vote Counter's secret key and the Administrator's secret key respectively.

$y_C = g^{x_C}(mod\ p)$: Vote Counter's public key

$y_A = g^{x_A}(mod\ p)$: Administrator's public key

H(.) : a cryptographically secure one way hash function

|| : which denote the concatenation of two strings

v_w : voting warrant

x_V : voter V_i's private key

First the VC goes for a hand shake with the administrator (A) and A gets the key for signing.

VC randomly selects $\bar{k} \in Z_q^*$ and computes,

$$r = g^{\bar{k}} \tag{5.1}$$

$$s = x_C + \bar{k}.H(v_w||r)\ (mod\ q) \tag{5.2}$$

VC sends (r, s) along with the voting warrant v_w to A via a secure channel.

Then, after receiving (r,s), A verifies the equation

$$g^s = y_C r^{H(v_w||r)}(mod\ p) \tag{5.3}$$

If it is correct, A accepts and computes,

$$s_{pr} = s + x_A y_C \tag{5.4}$$

As the key for signing the ballot of the voters.

A randomly select an integer $k \in Z_q^*$, and computes

$$t = g^{k+x_A}\ (mod\ p) \tag{5.5}$$

A then sends (r, t) to the voter V_i

Then, V_i selects two random numbers $a, b \in Z_q^*$

Voter V_i computes

$$r' = tg^a y_{pr}^b \ (mod\ p) \tag{5.6}$$

where, $y_{pr} = g^{s_{pr}} \ (mod\ p)$

$$e = H(r'||m) \ (mod\ q) \tag{5.7}$$

$$e^* = b - e \ (mod\ q) \tag{5.8}$$

If r'=0, then voter V_i needs to select a new tuple (a,b). Otherwise, voter V_i sends e^* to A.

After receiving e^*, A computes

$$s' = k + e^* s_{pr} \tag{5.9}$$

as the signed ballot and sends it to voter V_i.

After receiving s' from A, V_i computes

$$s^* = g^{u+s'} \ (mod\ p) \tag{5.10}$$

Thus, the signature on voting ballot m becomes finally (m, v_w, s^*, e).

Counting

Encrypting with VC's public key, V_i sends $(m, v_w, s^*, e) \ || \ NV_i)$ to VC.

VC verifies,

$$e = H(s^* y_B y_{pr}^e \ || \ m) \ (mod\ q) \tag{5.11}$$

If it is satisfied, the vote is accepted and final result is declared after the voting deadline is over.

5.3.3 Analysis of the Proposed Scheme

Completeness

The attacker cannot vote as a legal voter because during registration the voter sends encrypted message to RA using his own private key. Again, in the counting phase VC checks the signature from the administrator with the ballot. So, only authorized voters can participate in the voting process.

Uniqueness

Since RA issue a unique serial number to each legal voter only once, no voter cannot vote twice. RA and VC can detect the duplicate votes from that unique number.

Mobility

In this scheme the voter is not limited to voting in a particular voting booth. A voter can vote through the internet.

Anonymity

Administrator signs the blind ballot and the voted ballot is sent in an anonymous channel to vote counter. Hence the proposed scheme confirms this requirement.

Convenience

The proposed scheme does not require any additional requirement or does not need any extra skills. Hence it is convenience.

Fairness

Only after the deadline VC publishes the result. So, no one can not get it early.

5.4 Summary

With the rapid development of internet technology, voting through internet is a practical idea. In this scheme, a secure and efficient mechanism of electronic voting is proposed using the proxy blind signature. It increases the security of the voting system and also the impartiality factor is taken care. Hence, the proposed scheme can be practically applied in large scale voting.

Chapter 6

Conclusion

This thesis introduces a time stamped proxy blind signature scheme based on discrete logarithm problem(DLP) with the termination of delegation power. Proposed scheme satisfies all the security requirements of a proxy blind signature: distinguishability, nonrepudiation, unforgeability, verifiability, identifiability, prevention of misuse, unlinkability. When an abuse of a proxy is conducted in the proposed scheme, an original signer can identify the deviating proxy signer and terminate the abused proxies before the specified delegation time. Therefore, this scheme is suitable for many applications where the user's privacy and proxy signature are required.

Bibliography

[1] David Chaum. Blind signatures for untraceable payments. In *Crypto*, volume 82, page 199203, 1982.

[2] 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.

[3] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto. Proxy signatures: Delegation of the power to sign messages. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 79(9):1338–1354, 1996.

[4] Byoungcheon Lee, Heesun Kim, and Kwangjo Kim. Strong proxy signature and its applications. In *Proc of SCIS*, volume 1, pages 603–608, 2001.

[5] David Chaum. One-show blind signature systems, January 22 1991. US Patent 4,987,593.

[6] WD Lin and JK Jan. A security personal learning tools using a proxy blind signature scheme. In *Proceedings of International Conference on Chinese Language Computing*, pages 273–277. Illinois, USA, 2000.

[7] Tan Zuo-Wen, Liu Zhuo-jun, and Tang Chun-Ming. A proxy blind signature scheme based on dlp. 2003.

[8] Sunder Lal and Amit Kumar Awasthi. Proxy blind signature scheme. *Journal of Information Science and Engineering. Cryptology ePrint Archive, Report*, 72, 2003.

[9] Hung-Min Sun, Bin-Tsan Hsieh, and Shin-Mu Tseng. On the security of some proxy blind signature schemes. *Journal of systems and software*, 74(3):297–302, 2005.

[10] Wang Shu-hong, Wang Gui-lin, Bao Feng, and Wang Jie. Cryptananlysis of a proxy blind signature scheme based on dlp. 2005.

[11] Shaobin Wang, Hong Fan, and Guohua Cui. A proxy blind signature schemes based dlp and applying in e-voting. In *Proceedings of the 7th international conference on Electronic commerce*, pages 641–645. ACM, 2005.

[12] Qingshui Xue and Zhenfu Cao. A new proxy blind signature scheme with warrant. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 2, pages 1386–1391. IEEE, 2004.

[13] JG Li, YC Zhang, and ST Yang. Cryptanalysis of new proxy blind signature scheme with warrant. In *International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2005)*, 2005.

[14] Fuw-Yi Yang and Zhen-Wei Liu. Improvement of an efficient proxy blind signature scheme. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, pages 733–736. IEEE, 2009.

[15] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights. 2003.

[16] Yang Huaiqing and Wang Shaobin. A new e-voting scheme based on improved dlp. In *e-Business and Information System Security (EBISS), 2010 2nd International Conference on*, pages 1–4. IEEE, 2010.

[17] Samaneh Mashhadi. A novel secure self proxy signature scheme. *International Journal of Network Security*, 14(1):22–26, 2012.

[18] Guilin Wang. Designated-verifier proxy signature schemes. In *Security and privacy in the age of ubiquitous computing*, pages 409–423. Springer, 2005.

[19] Binayak Kar, Pritam Prava Sahoo, and Ashok Kumar Das. A secure proxy blind signature scheme based on dlp. In *Multimedia Information Networking and Security (MINES), 2010 International Conference on*, pages 477–480. IEEE, 2010.

[20] Jingfeng Su and Juxia Liu. A proxy blind signature scheme based on dlp. In *Internet Technology and Applications, 2010 International Conference on*, pages 1–4. IEEE, 2010.

[21] Young-Seol Kim and Jik-Hyun Chang. Provably secure proxy blind signature scheme. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pages 998–1003. IEEE, 2006.

[22] Wang Shaobin, Hong Fan, and Cui Guohua. Secure efficient proxy blind signature schemes based dlp. In *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, pages 452–455. IEEE, 2005.

[23] Rufen Huang and Qiang Nong. Security analysis of a proxy blind signature and its improved scheme. In *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, volume 2, pages 179–182. IEEE, 2011.

[24] Jianhong Zhang, Yuanbo Cui, and Zhipeng Chen. Security of proxy blind signature scheme. *Information Technology Journal*, 12, 2012.

[25] Eric Jui-Lin Lu and Cheng-Jian Huang. A time-stamping proxy signature scheme using time-stamping service. *International Journal of Network Security*, 2(1):43–51, 2006.

[26] GO Ofori-Dwumfuo and E Paatey. The design of an electronic voting system. *Research Journal of Information Technology*, 3(2):91–98, 2011.

[27] Jiguo Li and Shuhong Wang. New efficient proxy blind signature scheme using verifiable self-certified public key. *International Journal of Network Security*, 4(2):193–200, 2007.

[28] YU Baozheng and XU Congwei. A proxy blind signature scheme based on dlp. *Wuhan University Journal of Natural Sciences*, 12(1):83–86, 2007.

[29] Eric Jui-Lin Lu, Min-Shiang Hwang, and Cheng-Jian Huang. A new proxy signature scheme with revocation. *Applied mathematics and Computation*, 161(3):799–806, 2005.

[30] Manik Lal Das, Ashutosh Saxena, and Ved P Gulati. An efficient proxy signature scheme with revocation. *Informatica*, 15(4):455–464, 2004.

[31] Fuw-Yi Yang and Ling-Ren Liang. A proxy partially blind signature scheme with proxy revocation. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, pages 389–394. IEEE, 2010.

[32] Chin-Chen Chang and Jung-San Lee. An anonymous voting mechanism based on the key exchange protocol. *Computers & Security*, 25(4):307–314, 2006.

[33] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *Advances in CryptologyAUSCRYPT'92*, pages 244–251. Springer, 1993.

[34] Xuan Yang and Zhaoping Yu. An efficient proxy blind signature scheme based on dlp. In *Embedded Software and Systems, 2008. ICESS'08. International Conference on*, pages 163–166. IEEE, 2008.

[35] Tan Zhan and Yong-ping Zhang. Cryptanalysis of two proxy blind signatures based on dlp. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 6, pages V6–547. IEEE, 2010.

Dissemination

1. S.Panda and R.K Mohapatra, "Stamped Proxy Blind Signature Scheme ", International Journal of Computer Applications 64(15), February 2013

Accepted

2. S Panda and R.K Mohapatra, "An Application of Time Stamped Proxy Blind Signature in E-voting ", Second International Conference on Advances In Electronics, Electrical And Computer Engineering, Dehradun, India, 2013