

# Untraceable Blind Multisignature

**Ankit Kumar Namdeo**



**Department of Computer Science and Engineering  
National Institute of Technology Rourkela  
Rourkela-769 008, Odisha, India  
May 2015**

# **Untraceable Blind Multisignature**

*Thesis submitted in partial fulfillment of the requirements for the degree of*

## **Master of Technology**

*in*

## **Computer Science and Engineering**

**(Specialization: Information Security)**

*by*

**Ankit Kumar Namdeo**

**(Roll Number - 213CS2165)**

*Under the supervision of*

**Dr. Sujata Mohanty**



**Department of Computer Science and Engineering  
National Institute of Technology Rourkela  
Rourkela, Odisha, 769 008, India  
May 2015**

*Dedicated to my Parents and Siblings*



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

## Certificate

This is to certify that the work in this thesis entitled ” *Untraceable Blind Multisignature* ” submitted by *Ankit Kumar Namdeo* is a record of an unique research work did by him under our 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 specialization in **Information Security**, National Institute of Technology, Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT, Rourkela-769008

**Dr. Sujata Mohanty**  
Assistant Professor  
Department of CSE  
National Institute of Technology  
Rourkela-769008

## **Author's Declaration**

I hereby declare that all work contained in this report is my own work unless otherwise acknowledged. Also, all of my work has not been submitted for any academic degree. All sources of quoted information has been acknowledged by means of appropriate reference.

*Ankit Kumar Namdeo*  
*Roll No.: 213CS2165*  
*Department of Computer Science*  
*NIT Rourkela*

## Acknowledgment

As a matter of first importance, I might want to express my profound feeling of appreciation and appreciation towards my supervisor Prof Sujata Mohanty, who has been the guiding force behind this work. I need to say thanks to her for acquainting me with the field of Digital Signature and giving me the chance to work under her. Her unified confidence in this topic and ability to bring out the best of expository and handy aptitudes skills in people has been precious in intense periods. Without her important exhortation and help it would not have been feasible for me to finish this thesis. I am enormously obligated to her for her steady consolation and important guidance in every part of my academic life. I think of it as my favorable luck to have got a chance to work with such a sublime individual.

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

Amid my learns at N.I.T. Rourkela, I made numerous companions. I might want to express gratitude toward all of them, for all the colossal minutes I had with them.

When I glance back at my achievements in life, I can see a reasonable hint of my family's worries and commitment all over the place. My dearest mother, whom I owe all that I have accomplished and whatever I have turn into; my cherished father, for continually putting stock in me and rousing me to think beyond practical boundaries even at the hardest snippets of my life; and my sister; who was forever my quiet backing amid all the hardships of this attempt and past.

*Ankit Kumar Namdeo*

## Abstract

Multisignature is a variant of digital signature which enables a document to be signed by multiple signers simultaneously in a collaboration. It ensures the fairness property of the signer. Blind signature is another variant of digital signature in which a message is signed without disclosing its content. Blindness is an important property of blind signature in which, the message and the signature are unlinkable after signature is attached to the message.

In this Thesis, we designed a Blind Multisignature protocol with security features of blind signatures and multisignature. The security of the scheme lies in hard computational assumptions such as Integer Factorization problem (IFP), computational Diffie-Hellman problem (CDHP) and discrete logarithmic problem (DLP). The correctness of the scheme is tested mathematically and the scheme is also implemented in Java platform. The computational cost of the proposed scheme is low and the signature length (in byte) is nominal with the message size. The time of computation of each phase is computed and found to be low as compared to competent schemes. The security analysis of the scheme is done rigorously and the security features such as untraceability, blindness and unforgeability of the proposed scheme has been analysed and found secure under the attack. The scheme has properties of both blind signature and multi-signature. This scheme can be applied to real life applications such as electronic cash and electronic voting.

# Contents

<b>Certificate</b>	<b>4</b>
<b>Acknowledgment</b>	<b>6</b>
<b>Abstract</b>	<b>7</b>
<b>List of Figures</b>	<b>12</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Framework of Blind Signature . . . . .	1
1.2 Framework of Multisignature . . . . .	4
1.3 Security Features of Blind Multisignature . . . . .	5
1.4 Applications of Blind Multisignature . . . . .	6
1.4.1 Online Election System . . . . .	6
1.4.2 Digital cash . . . . .	6
1.5 Motivation . . . . .	6
1.6 Objective . . . . .	7
1.7 Contribution . . . . .	7
1.8 Organization of Thesis . . . . .	7
<b>2 Preliminaries</b>	<b>9</b>
2.1 Cryptography Concepts and Digital Signature . . . . .	9
2.1.1 Digital Signature . . . . .	10
2.2 Mathematics of Cryptography . . . . .	10



2.2.1	Prime Numbers and Primality Testing . . . . .	10
2.2.2	Miller Rabin Primality Test . . . . .	11
2.2.3	Generation of Prime Numbers . . . . .	11
2.2.4	Hash Function . . . . .	11
2.2.5	Integer factorization Problem . . . . .	12
2.2.6	Discrete Logarithmic Problem . . . . .	12
<b>3</b>	<b>Literature Review</b>	<b>14</b>
<b>4</b>	<b>Untraceable Blind Multisignature</b>	<b>16</b>
4.1	Proposed scheme . . . . .	16
<b>5</b>	<b>Security Analysis of Proposed Algorithm</b>	<b>20</b>
5.1	Blindness . . . . .	20
5.2	Untraceability . . . . .	21
5.3	Unforgeability . . . . .	21
5.4	Correctness . . . . .	22
<b>6</b>	<b>Implementation and result</b>	<b>23</b>
6.1	Implementation . . . . .	23
6.2	Results . . . . .	24
<b>7</b>	<b>Conclusions and Future Work</b>	<b>28</b>

## List of Abbreviations

<b>BS</b> .....	Blind Signature
<b>IND CCA1</b> ...	Indistinguishability Chosen Ciphertext Attack
<b>IND CCA2</b> ...	Indistinguishability Adaptive Chosen Ciphertext Attack
<b>DLP</b> .....	Discrete Logarithmic Problem
<b>IFP</b> .....	Integer Factorization Problem
<b>CDH</b> .....	Computational Diffie-Hellman Problem
<b>DDH</b> .....	Decision Diffie-Hellman Problem
<b>MS</b> .....	Multisignature
<b>BMS</b> .....	Blind Multisignature

# List of Acronyms

---

Acronym	Description
$U_i$	Individual Signer
$x_i$	Individual private key of signer
$y_i$	Individual public key of signer
$Y$	Group public key
$g$	generator of a group
$k_i$	Random number chosen by the signer
$R_i$	$R_i$ is the value calculated by each individual signer using $k_i$
$V$	Product of all the $R_i$
$\alpha$	Random variable chosen by the Requester
$\beta$	Random variable chosen by the Requester
$M$	Message of the Requester
$M'$	Blinded Message
$R$	$R$ is the variable calculated by using $V, \alpha, \beta$
$S_i$	Sign done by individual signer
$J$	Summation of all the signs
$S$	Extracted signature

---

# List of Figures

1.1	Blind Signature . . . . .	4
1.2	Multisignature . . . . .	5
4.1	Block Diagram of the key generation phase . . . . .	19
6.1	Output of Blind multisignature part 1 . . . . .	26
6.2	Output of Blind multisignature part 2 . . . . .	27

# Chapter 1

## Introduction

### 1.1 Framework of Blind Signature

The first scholar to propose the concept of the blind signature scheme was D. Chaum is the first one who bring the concept of blind signature in 1982[12]. The anonymity of the members is guranteed in the blind signature scheme. There are two parties in this signature scheme the requester R and the signer S[3]. Message M will be signed by the signer and send to the requester which wants the sign on that message. At the begining the Requester blinds the message M into M' and sends M' to S. Then S signs the message M' and output the s' to the requester R. After recieving the s' the requester R unblinds the signature s' to s which is the signature on the message M. the content of message M can be protected by R in this scheme. Likewise, at whatever point S doles out a mark pair of (M,s), B can't focus when, or for whom he/she marked that message. A few applications which utilize blind signature are internet voting and digital cash. When we present an online vote, we may like for that vote to be unknown so nobody can tell whom we voted in favor of. Also with electronic money, we may not need another person to know who we are the point at which we spend it. This is like ordinary paper money, when we make a purchase,the seller pretty much has no clue who we are, yet we can presumably tell whether the cash we issued him is legitimate. Specically, in this electronic money situation, a report relates to an electronic coin or note, and the endorser speaks to a bank.The high-roller holds namelessness in any exchange that includes electronic coins on the off chance that they are signed blindly.

**Variations of Blind Signature Restrictive Blind Signature:**

Restrictive blind signature means that a requester can blind the documents but with some restrictions. It is a protocol which says that any user can request for a blind signature on a document from a valid signer. But it has certain limitations as compared to the normal blind signature[28]. Like normal blind signature the user can blind the message in any way but with the restriction in the choice of message and should follow the certain protocols so that the genuine message and the blinded message are isomorphic[28]. The blind signature ensures that the signature generated by the signer for one transaction can only be used once. But if the requester becomes malicious and tries to replay the signature again after some time duration then the identity of the requester should be revealed. This can be done by applying restrictive blindness to the normal blind signature scheme.

**Revocable Anonymity:**

In any communication, protecting the contents is not enough. Sometimes it is required to keep the identity of the recipient as private. In the context of electronic commerce, if no anonymity is provided then the users preferences can be known. With this information anyone can know the profile of users and send them targeted advertisements or can sell the profiles to other commercial units. The buyer will get problem by this as they want to do the transactions anonymously. Blind signature allows a user to do any transactions anonymously. But in case of any legal disputes the identity of the malevolent user need to be revealed. This is known as revocable anonymity i.e to revoke the anonymity when needed[13].

**Fair Blind Signature:**

Though it is another variation of blind signature, it can be obtained from the restrictive blind signature also[23]. In a fair blind signature protocol a single trustee or multiple trustees may get involved in the system. It is also used to revoke the anonymity of malicious users and the trustee used to do that. To do so, the trustee view all the parts of the blinding process. For this reason the trustee need to be remain online all the time, which compromises the efficiency of the system. Later many fair blind signatures are developed in which the trustee need to keep a public-private key pair. The trustee can only involved in the tracing protocol and by using the key pairs he can trace the identity of the malicious

user.

### **Partial Blind Signature:**

To achieve revocable anonymity, another variation of blind signature called as partial blind signature is also used. To trace the identity of the malicious user, the signer need to keep some data in the database during the transaction. This will increase the space of the database. When the requester tries to use the signature twice, the signer checks the database to identify that requester. But to search the database each time is not so feasible. Partial blind signature overcomes this problem[3]. In a partial blind signature protocol, the signer and the requester have some common agreed information. The requester can blind the message but the common agreed information need to be remain unblind. By using the common information the signer can trace the identity of the requester when needed. The concept of partial blind signature was developed by Abe and Okamoto[3]. The blind signature has 4 phases they are:

1. **Blinding Phase:** In this phase the requester hides the message or blind the message for the signer such that the signer can not be able to see the actual content of the message and he did that by either multiplying the message with random number or by encrypting it with some key or it can be hashed also.
2. **Signing Phase:** In this phase the signer signs the message by its own signature but without revealing the actual content of the message. The signer signs blindly on the message sent to him by the requester.
3. **Unblinding Phase:** In this Phase the requester unblind the message sent by the signer.
4. **Verifying Phase:** In this phase the verifier receives the signature and it verifies the legitimacy of the signature by checking the verifying equations.

The following requirements, namely, correctness, blindness, unforgeability and untraceability must meet for blind signature.

- **Correctness:** Those who has the signer's public key will be able to identify the signature of the message signed by using the blind signature scheme.

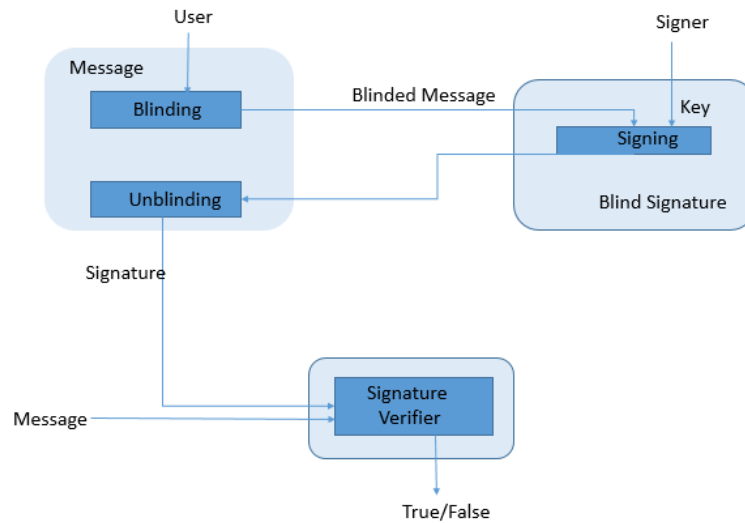


Fig. No. 1 Blind Signature Flow Chart

Figure 1.1: Blind Signature

- **Blindness:** The signer must not be able to find the content of the message.
- **Unforgeability:** No one else beside signer will be able to derive any forged signature and pass the verification process because the signature is the proof of the signer[15].
- **Untraceability:** There must be no link between the message and the signature and no one including the signer will be able to find it[14].

## 1.2 Framework of Multisignature

A multisignature scheme is a variant of digital signature scheme, which enables a document to be signed by multiple signers simultaneously in a collaboration[18]. Normal signature scheme is used by all the signers for signing a document. Multisignature is more secure and eliminates the latest attacks. Individual signers are identified by the Information contained in Multisignature. The main drawback of this scheme is that both the length of the signature and the computation cost of its computation for verification increases linearly according to the number of signers. Two important properties which must be fulfilled by the multisignature to achieve the optimum signature are[20]:

- Individual signature and multisignature must be of the same size.
- Individual signature and multisignature must use the same verification process.



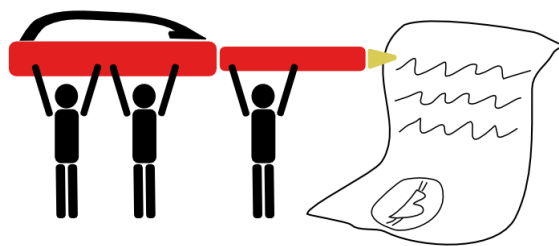


Figure 1.2: Multisignature

### 1.3 Security Features of Blind Multisignature

The following are some of the security features in Blind Multisignature:

**Unforgeability:** No one else beside signer will be able to derive any forged signature and pass the verification process because the signature is the proof of the signer[15].

**Untraceability:** There must be no link between the message and the signature and no one including the signer will be able to find it[14].

**Blindness:** The Signer should not be able to know to find the content of the message even if he has the message signature pair.

**IND-CCA1 attack:** Indistinguishability is an important property in cryptosystem, if the cipher text is indistinguishable then it will be hard for the adversary to identify the different pair of cipher text based on the encrypted message. Indistinguishability is the basic requirement. Semantic security and IND-CCA are equivalent and are being used interchangeably in many cryptographic proofs. IND-CCA attacks can also be performed on digital signatures[4].

The attacker send two messages to the challenger then the challenger take one message randomly and encrypt it and resend to the attacker, now a cryptosystem is considered secure in terms of indistinguishability if he is not able to find which message is encrypted with probability greater than  $1/2$ . If attacker is able to identify the message with probability greater than  $1/2$ , then attacker has the advantage in identifying the encrypted message, and the scheme is not secure in terms of indistinguishability [9].

## 1.4 Applications of Blind Multisignature

### 1.4.1 Online Election System

E-voting is a most critical utilization Blind Multisignature scheme. Voter is free from of any reasonable in light of the fact that he/she put make their choice indiscriminately administrator is only the power who gives the sign. E-voting application may be sorted out by any administration delegate, private association, or any extraordinary gathering of individuals. The security of client who make the choice is keeping hidden. Each client's make choice can be effectively confirmed with the assistance of administrator's identity. The privacy issue identified with digital signature is a touch explained by Blind Multisignature scheme[22].

### 1.4.2 Digital cash

Digital cash or e-cash was first introduced by D. Chaum as an anonymous cash system. It is interesting to know that ecoins are blind signatures.

So we can see one transaction can give one valid token packet or one valid signature. For multiple transaction the corresponding signatures or the e-coins will be different. But, nowadays many requester becomes malicious and spends the e-coins for multiple times. This is known as the double spending problem. Though blind signature provides untraceability or unlinkability but sometimes it is necessary to reveal the identity of the requester. To do so, one requester should not blind all the internal structure of the message. It should blind the outer part of the message so that by using the public parameters the signer can able to trace the identity of the malicious requester. This is kind of blind signature is known as restrictive blind signature[21].

## 1.5 Motivation

In e-commerce world such as an e-voting system authentication is very important as well as anonymity and confidentiality. The answer to this problem is a blind multisignature that is based upon a difficult trapdoor function which is discrete logarithm problem. So we intend to design such a scheme that is immune against most of the common cryptographic attacks and along with the security low computational overhead also.

## 1.6 Objective

To create another Blind Multisignature scheme based upon computationally hard suspicion like discrete logarithmic problem (DLP). The main focus of Our scheme will fundamentally concentrate on the following attributes:

1. To design a Blind multisignature with the properties of blind signature as well as multisignature
2. It should have low computational cost and computational overhead.
3. It must fulfil all the requirements namely correctness, unforgeability, unlinkability, and blindness.

## 1.7 Contribution

In this Thesis, we designed a Blind Multisignature protocol with security features of blind signatures and multisignature. The security of the scheme lies in hard computational assumptions such as Integer Factorization problem (IFP), computational Diffie-Hellman problem (CDHP) and discrete logarithmic problem (DLP). The correctness of the scheme is tested mathematically and the scheme is also implemented in Java platform. The computational cost of the proposed scheme is low and the signature length (in byte) is nominal with the message size. The time of computation of each phase is computed and found to be low as compared to competent schemes. The security analysis of the scheme is done rigorously and the security features such as untraceability, blindness and unforgeability has been analysed and found secure under the attack. The scheme has the properties of blind signature anonymity as well as multi-signature. This scheme can be applied to real life applications such as electronic cash and electronic voting.

## 1.8 Organization of Thesis

The rest of Thesis is organized as follows:

**Chapter 2:**In this chapter, we have discuss some of the basic functionalities and basic concepts which plays an vital role in the proposed work. In order to understand the functioning of the proposed algorithm, the reader must go through these preliminaries to have

a better understanding of the work. proposed.

**Chapter 3:**In this chapter, we have studied some papers related to the blind multisignature scheme.

**Chapter 4:**In this chapter, we have presented our proposed work about the new untraceable blind multisignature scheme.

**Chapter 5:**In this chapter, we analyzed the security measures of our proposed scheme.

**Chapter 6:**In this chapter, we presented the results of the program which is implemented in java.

**Chapter 7:**In this chapter, We put some light on the future work and also concluded our work.

# Chapter 2

## Preliminaries

In this chapter, we retrospect the literature related to digital signature, mathematics of cryptography and hash functions. First, we give a brief overview of cryptography concepts and digital signatures. In the middle of this chapter we discuss about the prime numbers and primality test, we also write about how the random numbers are generated, how prime numbers are generated and why hash functions are important in cryptography. At the last we discuss preliminaries related to discrete logarithms and Integer factoriation.

### 2.1 Cryptography Concepts and Digital Signature

Cryptography can be characterized as ensuring data by changing into an indiscernible arrangement, known as cipher text[8].The cipher text can only be decrypted by those who has the secret key. Encoded messages can rarely be broken by cryptanalysis, moreover called code breaking, albeit present day cryptography strategies are for all intents and purposes unbreakable. Cryptography can be divided in to two categories, one which uses single key are known as symmetric key systems and the other which uses two keys are public key system. In symmetric key system the sender and receiver share the same secret key, whereas in public key cryptography the two keys are used one is public to all and another is private key which is only used by recipient of the messages[29]. In case of signature the signer sign the message with its own private key and the message will be decrypted by using signers public key ensuring its authenticity.

### 2.1.1 Digital Signature

The digitization of paperwork has been a major leap in the field of creation and transfer of documents[29]. Digital signature solves the major security concern for the document. It is being a digital analog of handwritten signatures and is crucial for identifying the the sender's identity and also whether the receiver has received it tamper free[8]. The services provided by digital signature are:

1. Message integrity
2. Non-repudiation
3. Authentication

But the big cons of digital signature come when the user needs to identify himself during transactions like purchase (other than cash) or obtaining a service. This breaches the privacy of the person in concern. Organizations now have massive amounts of data, threatening these users' security. Taking it forward, where a digital signature reveals the identity of the person in any transaction whereas a Blind signature protects the sender's privacy and enables the user to get a signature without giving the actual message to the signer.

## 2.2 Mathematics of Cryptography

### 2.2.1 Prime Numbers and Primality Testing

A primality test is an algorithm to find out that whether a given number is prime or not. A primality test only gives output whether a given number is prime or not in yes or no[8]. In factorization we have to find the factors of a number and in primality test we just have to check and to find factors we need more computation so it is computationally more troublesome than primality test. There are two types of primality test.

**Deterministic Algorithm:** In deterministic primality testing algorithm takes an input and deterministically produce an output whether a given number is prime or not.

**Probabilistic Algorithm:** In Probabilistic algorithm it takes an integer number as input and produce an output with some error that whether a number is prime or not. It cannot

tell deterministically the difference between composite number and prime number, but it is faster than deterministic algorithm[25].

### 2.2.2 Miller Rabin Primality Test

In the field of cryptography prime numbers are mostly required. Many methods are there to generate the prime numbers like Fermat's or Mersenne's or Safe prime method. But if at any instance, these methods have failed to create a prime number then problems will arise[29]. To overcome these problems, Cryptography provides many primality testing methods. One of the methods that we have used in our implementation part is Miller Rabin's primality test. Miller Rabin method is a probabilistic algorithm. Miller Rabin primality test is the combination two other probabilistic methods which are Fermat test and Square root test. In this method we write  $n - 1$  as the product of an odd number  $m$  and a power of two.  $n - 1 = m^k$ . As we know, the Fermat test in base  $a$  can be written as  $a^{n-1} = am^k = a^{[m]}$  In the above step instead of calculating  $a^{n-1} \pmod n$  in one step, we are doing it in  $k + 1$  steps. The benefit is square root test is performed in each step. If at any step the square root test fails to satisfy then we declare the number as composite[8].

### 2.2.3 Generation of Prime Numbers

For generating prime numbers we have used Mersenne Prime method. It has the formula  $M_p = 2^p - 1$ . As per the formula if  $p$  is a prime number then  $M_p$  was thought to be prime.

### 2.2.4 Hash Function

We need the one way hash function to generate the message digest of the message. The message and the message digest is equivalent to a document and the corresponding finger print. We calculate the message digest in order to achieve message integrity[8]. To create the message digest the message is passed through a cryptographic hash function. There are many hash functions designed by Ron Rivest. These hash functions are used to create the message digest. These are referred to as MD2, MD4, MD5. MD stands for message digest. We have used the MD5 hash function to create the message digest[29]. MD5

takes the message as the input and divides the message into blocks of 512 bits and creates a digest of 128 bits.

### 2.2.5 Integer factorization Problem

Integer factoring means the composite number is decomposed into the multiples of smaller integers. In prime factorization we bound those integers to be prime numbers. There is now efficient algorithm for very large numbers for integer factorization. Numerous zones of science, mathematics and computer engineering have been presented as a powerful influence for the issue, including elliptic curves, logarithmic number hypothesis and quantum computing.

### 2.2.6 Discrete Logarithmic Problem

Discrete logarithms were used mainly in computations of finite fields and elliptic[2]. Discrete logarithm problem has significant importance in the field of cryptography as the complexity lies in solving the discrete logarithm problem[1]. In case factorization problem, the security of the whole system lies on a single number  $n$ . If the attacker can factorize the number  $n$  then it will break the security of the system[27]. Whereas, a discrete logarithm problem says it is very easy to compute  $a = g^x$  given  $x$  and  $g$ , where  $g$  is the public parameter and  $x$  is the private parameter, but it is very difficult to compute  $x$ , given  $a$  and  $g$ , which are public parameters. Here  $g$  is the primitive element and it is the element of a cyclic finite group. Let  $G(q)$  is a group and  $G(q)^*$  is the multiplicative subgroup in which all the elements are having their multiplicative inverse. Here  $q$  is a prime number. An element  $g$  is called as primitive element such that  $g \in G(q)$  and it generates the cyclic multiplicative subgroup  $G(q)^*$  of the group  $G(q)$ . Any element  $a \in G(q)^* = G(q) - 0$ , the discrete logarithm of  $a$  with respect to  $g$  is that integer  $x$ ,  $0 \leq x \leq q - 1$ , for which  $a = g^x$ . Here  $x = \log_g^a$ . The DLP is very easy to implement and it is used mostly in Ecash system.

The discrete logarithm issue has gotten much consideration lately; portrayals of probably the most productive calculations for discrete logarithms over limited fields can be found in numerous calculation. The best discrete logarithm calculations have anticipated that running times comparative would those of the best considering calculations. Rivest



has investigated the normal time to take care of the discrete logarithm issue both regarding figuring power and expense.

As a rule, the discrete logarithm in a self-assertive gathering of size  $n$  can be figured in running time  $O(\sqrt{n})$ , however in numerous gatherings it should be possible speedier.

In similar to the factoring problem, the DLP is accepted to be troublesome furthermore to be the hard heading of a restricted capacity. Hence, it has been the premise of a few open key cryptosystems, including the ElGamal framework and DSS. The DLP bears the same connection to these frameworks as considering does to the RSA framework: the security of these frameworks lays on the suspicion that discrete logarithms are hard to figure. Despite the fact that the DLP exists in any gathering, when utilized for cryptographic purposes the gathering is normally  $Z_n^*$ [16].

# Chapter 3

## Literature Review

Blind multisignature is the combination of blind signature and multisignature so it has the properties of both. It provides anonymity of the blind signature and fairness property of the multisignature.

Patrick Horster, Markus Michels and Holger Peterson[22] present the first blind multisignature scheme based on the discrete logarithm problem. The advantage of the scheme is that it gets rid of the assumption that all communication must be written on a public board (to be more precise, the encrypted vote and, later, the vote itself and the additional parameter must be written on the board) and gets rid of the additional communicating phase to open the commitment. In this scheme we have to assume that at least one of the administrators is honest.

There are two arguments that this assumption is reasonable: First, by use in practice, the anonymous channel would be simulated by a mix-net where it is assumed that at least one mix-center is honest as well. Therefore, a trustworthy entity must be assumed anyway.

Second, in the initialization of the scheme, system parameters without trapdoors must be chosen by the administrators or other authorities. For example, Chen and Burmester assumed the existence of a trusted center to generate a composite module which is needed to use the Fiat-Shamir scheme in their system. Clearly, if this center is untrustworthy, the security is completely lost. While it is difficult in this case to distribute this center into several centers where only one is honest, it seems to be possible if the security of the used schemes are based on the discrete logarithm problem[11]. One honest center can avoid that, say, a trapdoor-prime or a trapdoor-generator is chosen, and can guarantee that public keys of the administrators are authentic. Obviously, the centers' tasks can be done

by the administrators, if atleast one is honest. Then, the existence of a trusted center is not necessary. As a result, it seems reasonable to assume that at least one administrator is honest.

A more serious problem of the schemes mentioned so far, is the possibility of verifiable buying of votes. The coercer, who taps the line between the administrators and the voter, might force the voter to use random numbers prepared by him in the voting slip issuing phase. Then he sees in the voting phase if these numbers will appear again or not and therefore check if the voter votes the "correct" candidate. This might be prevented physically if the voter can't determine the random numbers by himself. If he is supplied by random numbers by a (trusted) physical device, which also does the computation for him, then this attack will fail. Clearly, a more powerful coercer, who can physically see what the voter votes, can still be successful. This coercer model, however, seems to be less of practical than of theoretical interest as the effect of vote-buying is only non negligible in large scale elections, if the number of bought votes is high. As the powerful coercer can't see the vote of several voter simultaneously, he needs a large number of supporters for supervising the voters. This scenario seems not to be very realistic.

# Chapter 4

## Untraceable Blind Multisignature

### 4.1 Proposed scheme

The proposed scheme consists of three participants, namely, a group of signers, a trusted third party, and a requester. It consists of five phases: key generation, blinding, signing, unblinding, and verification.

Suppose,  $U$  be the group of signers such that  $U = \{U_1, U_2, .. U_n\}$ . Each member  $U_i$  is responsible for signing message  $M$ . Let there be a group of signers be  $U_1, U_2, .. U_n$  and the message  $M$ . A trusted third party(TTP) decides a large prime number  $p$ , and a prime divisor  $q$  such that  $q|(p - 1)$  and a one way hash function  $h$ .

The operation of each phase is described below.

#### *Key generation :*

Each signer  $U_i$  has to choose its own secret key  $x_i$  such that  $1 < x_i < q$ .  $g$  is the generator of cyclic group of order  $q \in Z_q^*$ . Each signer  $U_i$  disclose their public key  $y_i = g^{x_i} \pmod{q}$ . After every signer disclose their public key, TTP calculates combined(group) public key as follows,

$$Y = \prod_{i=1}^n y_i \pmod{q}. \quad (4.1)$$

#### *Blinding phase :*

Operation of blinding phase is as follows.

**Step 1:** Every signer  $U_i$  selects a random number  $k_i \in Z_q^*$  and compute  $R_i$  as follows.

$$R_i = g^{k_i} \pmod{q}. \quad (4.2)$$

Then sends  $R_i$  to the requester.

**Step 2:** After receiving  $R_i$  from all the signers, the requester chooses random number  $\alpha, \beta \in Z_q^*$  and computes,

$$V = \prod_{i=1}^n R_i \pmod{q}. \quad (4.3)$$

$$R = V^\alpha g^\beta \pmod{q}. \quad (4.4)$$

**Step 3:** The requester blinds the message as follows,

$$M' = \alpha M V R^{-1} \pmod{q}. \quad (4.5)$$

and sends the blinded message( $M'$ ) to the signer.

*Signing phase :*

In this phase a blinded message is to be signed by the signers.

**Step 1:** After receiving the blinded message each signer computes  $S_i$  as follows,

$$S_i = (k_i.M' + V.x_i) \pmod{q}. \quad (4.6)$$

Then the signer sends  $S_i$  to the TTP.

**Step 2:** After receiving all  $S_i$  from all the signers, the TP calculates the multisignature as follows,

$$J = \sum_{i=1}^n S_i \pmod{q}. \quad (4.7)$$

and sends to the requester.

*Unblinding phase :*

The requester unblinds the signature as follows,

$$S = (JRV^{-1} + \beta M) \pmod{q}. \quad (4.8)$$

Finally the requester get the message signature pair (M, S, r) where

$$r = R \pmod{q}. \quad (4.9)$$

*Verification :*

The verifier can verify the sign by the equation stated below. If the below equation satisfies then the signature is valid and legitimate.

$$T = (g^S y^{-V})^{M^{-1}} \pmod{q}. \quad (4.10)$$

$$r = T \pmod{q}. \quad (4.11)$$

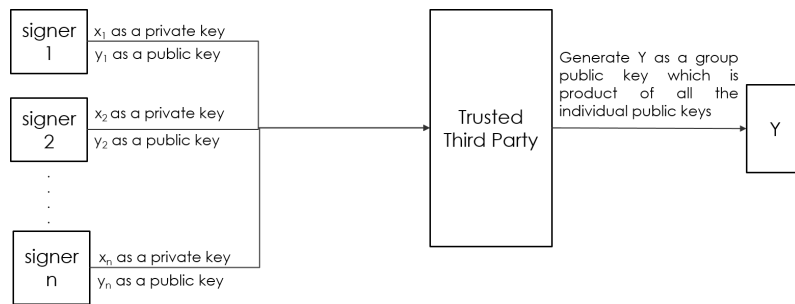


Figure 4.1: Block Diagram of the key generation phase

# Chapter 5

## Security Analysis of Proposed

### Algorithm

This section shows that this scheme preserves all the characteristic of a blind signature.

#### 5.1 Blindness

Blindness or unlinkability is the property , which prohibits the signer to link the blinded message to the original message. The signer signs the message without knowing what is the content of the message[7]. In this scheme, the requester calculates the blinded message as  $M' = \alpha M V R^{-1} \text{ mod } p$ . If any signer has the intention to see the content of the message before signing it he was unable to do so because  $\alpha$  and  $\beta$  are chosen randomly by the requester so it makes hard for the signer to reveal the contents of the message and we are using another factor as a multiple in the blindness equation i.e.  $V$  which is the summation of all the  $R_i$  values generated by each signer so it is almost insuperable for an individual signer to know the content of the message. Hence, the signer will not be able to see the message  $M$ .



## 5.2 Untraceability

Untraceability is the property of the blind signature acheme which let a signer unable to link the message and signature even though the signature is public.[19] If someone gets the valid signature, it is hard to link the signature to the message. In this scheme, if the signer keep arecord set  $(k_i, R'_i, M, S'_i, V, x_i)$ , where  $i= 1, 2 .. n$ , then also it is hard to trace the blind signature. When the requester discloses n records  $(M_i, R_i, S_i)$  to the public the signer will compute the values  $M, R, V$ . However, the signer will not be able to trace the blind signature by detecting whether each  $R_i$  and  $R_{i+1}$  have the same relation. Hence, it is hard to trace the signature in this scheme.

## 5.3 Unforgeability

Forging  $(M, S, r)$  is hard because the discrete logarithm problem is hard to solve. Assume two cases as follows.

**Case 1 :** If an adversary try to faux  $r_1, M_1$ , he will be unable to get  $S_1$ . Since

$$r = T \pmod q. \quad (5.1)$$

$$r_1 = (g^S y^{-r})^{M^{-1}}. \quad (5.2)$$

and  $S_1$  is unknown. This is a discrete logarithm problem and hard to solve.

**Case 2 :** If an adversary to faux  $M_1, S_1$ , he will be unable to get  $r_1$ . Since

$$r = T \pmod q \quad (5.3)$$

$$r_1 = (g^S y^{-r})^{M^{-1}} \quad (5.4)$$

and  $r_1$  is unknown. This is also a discrete logarithm problem and hard to solve.

## 5.4 Correctness

The correctness of the verification equation is shown below,

$$\begin{aligned}
T &= (g^S y^{-V})^{M^{-1}}. \\
r &= T \pmod{q}. \\
r &= (g^S y^{-r})^{M^{-1}} \\
&= g^{(JRV^{-1} + \beta M - (\sum_{i=1}^n x_i)r)}^{M^{-1}} \quad (5.5) \\
&= V^\alpha g^\beta \\
&= R \pmod{q}. \\
&= r
\end{aligned}$$

The proposed Blind signature scheme is based upon the security of solving hard computation assumption such as DLP and IFP. It is not possible to attack at this scheme to obtain private keys. The proposed scheme use complex function in order to obtain high security. Analysis of security features is done and found that it is resistant against forgery attack such as existential and selective forgery. Proposed blind multisignature scheme claims to be more secure than existing scheme. It is reliable for confidential transaction, e-commerce, e-cash, e-voting, communication etc.

# Chapter 6

## Implementation and result

### 6.1 Implementation

The Proposed Scheme is implemented in java platform. We use netbeans IDE 7.3 as integrated development environment. We don't need any database because we are not storing the keys in our algorithm. In our program we use java big integers for computing very large numbers. We use cryptography package and security package to generate random numbers and generators. We use the hash function in java to get the message digest, by using SHA-2 algorithm. The message size we chose is of 5KB.

The standard hardware configuration is :

1. Hard disk should be 90 GB
2. RAM 2GB.
3. OS can be of user's choice.

The implementation consists of following steps in the proposed scheme:

1. Key Setup
2. Blinding of the Message
3. Signature
4. Unblinding
5. Verification

## 6.2 Results

The proposed scheme is implemented with AMD quad core processor along with 4 GB RAM in using java. After execution we got the following time for each phase in our algorithm using ""System Time".The hardware is same for all the phases.

First of all generate a generator by taking input a large prime number. We also want private key of the signers. here in this code we use 5 signers.

value of generator g 5421644057436475141609648488325705128047428394380474376834667300766108262613900542681289080713724597310673074119355136085795982097390670890367185141189796.

value of p 132323768951986124075479307182674357577285270296234088722451560397577130290368719146452186041204237350521785240337048752071462798273003935646236777459223.

value of q 857393771208094202104259627990318636601332086981.

Enter private key Number : 46464

Enter private key Number : 313215

Enter private key Number : 648454

Enter private key Number : 34165

Enter private key Number : 654846

Public key for signer 1 828219181897331915706343176718742175216868673231

Public key for signer 2 762163890635232521565637106043926028507037525621

Public key for signer 3 781439396482122571608505562948684655492903274801

Public key for signer 4 807613618464235894124234821305274496437499862119

Public key for signer 5 790421985953475447588739851809956423193647769347

Hashing process for message

Taking input from file and converting it into the message digest

Hex format : 173d87aeb834951ef097585eb2550fed8653caf4d047759855c77f88aadb402c

Converting hexcode to biginteger for use in program 656994352159577377578393021552384875855680944817886350027289388127646072876

Verification process

The signature is verified using the verification algorithm and the result of verification is published as true/false. In this case the result is true.

The computational time for each phase are:

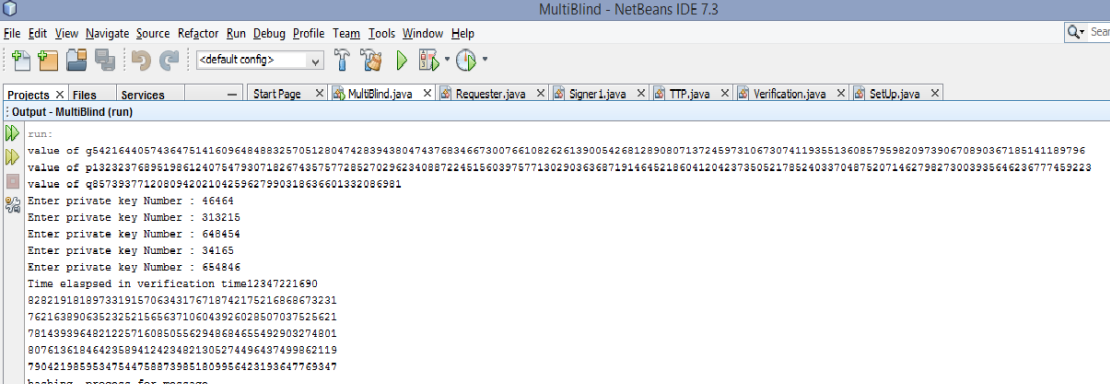
Time elapsed in Blinding time 4.03ms

Time elapsed in Signing time .015ms

Time elapsed in Unblinding time 1.05ms

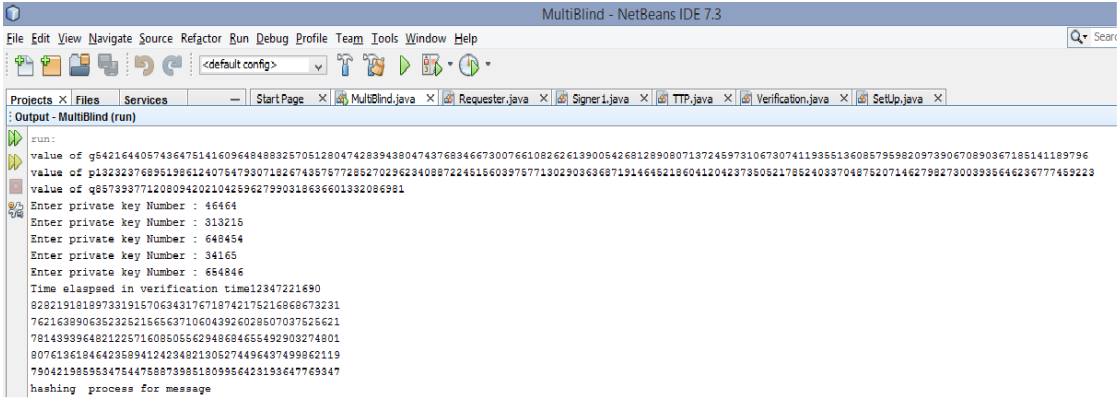
Time elapsed in Verification time .0129ms

The message length taken is of 5KB and the signature generated is of 20 Byte.



```
MultiBlind - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects X Files Services - StartPage X MultiBlind.java X Requester.java X Signer.java X TTP.java X Verification.java X SetUp.java X
Output - MultiBlind (run)
run:
value of g5421644057436475141609648488325705128047428394380474376834667300766108262613900542691289080713724597810678074119355136085795982097390670890367185141189796
value of p13232376895198612407547990718267435757728527029623408872245156039757713029036368719146462186041204237350521785240337048752071462798273003935646236777459223
value of q857393771208094202104259627990318636601332086991
Enter private key Number : 46464
Enter private key Number : 313215
Enter private key Number : 648454
Enter private key Number : 34165
Enter private key Number : 654846
Time elapsed in verification time12347221690
828219181897331915706343176718742175216868673231
762163890635232521566637106043926028507037525621
781439896482122571608505562948684655492903274801
807613618464235894124234821305274496437499862119
790421988953475447588733851809956423193647769347
hashing process for message
```

Figure 6.1: Output of Blind multisignature part 1



```
MultiBlind - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects X Files Services StartPage X MultiBlind.java X Requester.java X Signer.java X TTP.java X Verification.java X SetUp.java X
Output - MultiBlind (run)
run:
value of g5421644057436475141609648488325705128047428394380474376834667300766108262613900542691289080713724597810678074119355136085795982097390670890367185141189796
value of p132323768951986124075479907182674357577285270296234088722451560397577130290363687191464652186041204237350521785240337048752071462798273003935646236777459223
value of q857393771208094202104259627990318636601332086991
Enter private key Number : 46464
Enter private key Number : 313215
Enter private key Number : 648454
Enter private key Number : 34165
Enter private key Number : 654846
Time elapsed in verification time12347221690
828219181897331915706343176718742175216868673231
762163890635232521566637106043926028507037525621
781439896482122571608505562948684655492903274801
807613618464235894124234821305274496437499862119
790421988953475447588733851809956423193647769347
hashing process for message
```

Figure 6.2: Output of Blind multisignature part 2

# Chapter 7

## Conclusions and Future Work

The proposed BS scheme is based upon the hard computation assumption i.e. DLP. The proposed scheme is implemented in Java. It is also analysed and verified successfully. We had done the security analysis of our proposed scheme and found it resistant to DLP attacks. The proposed scheme can have wide range of application in areas such as e-cash, voting, e-commerce. It ensures to be more secure than existing scheme. The proposed scheme ensure, verifiability, non-repudiation, identityability. We are trying to make a more secure blind multi signature using ECDLP in future by improving the current scheme.



# Bibliography

- [1] Murat Ak, Turgut Hanoymak, and Ali Aydın Selçuk. Ind-cca secure encryption based on a zheng–seberry scheme. *Journal of Computational and Applied Mathematics*, 259:529–535, 2014.
- [2] Jan L Camenisch, Jean-Marc Piveteau, and Markus A Stadler. Blind signatures based on the discrete logarithm problem. In *Advances in Cryptology—EUROCRYPT’94*, pages 428–432. Springer, 1995.
- [3] Zhenfu Cao, Haojin Zhu, and Rongxing Lu. Provably secure robust threshold partial blind signature. *Science in China Series F: Information Sciences*, 49(5):604–615, 2006.
- [4] Hua Chen, Jianhua Chen, and Guangxing Cai. Cryptanalysis of a new blind signature based on the dlp. In *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, pages 415–418. IEEE, 2010.
- [5] Paul Chevalier, B Dorval, and C Menard. Random number generator, February 5 1974. US Patent 3,790,768.
- [6] Cécile Delerablée and David Pointcheval. Dynamic fully anonymous short group signatures. In *Progress in Cryptology-VIETCRYPT 2006*, pages 193–210. Springer, 2006.
- [7] Xiaobo Feng and Mingqiang Wang. Attack on the cryptosystem based on dlp. In *CIS*, pages 896–899, 2011.
- [8] Behrouz A Forouzan. *Cryptography & Network Security*. McGraw-Hill, Inc., 2007.

- [9] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
- [10] Yoshikazu Hanatani, Yuichi Komano, Kazuo Ohta, and Noboru Kunihiro. Provably secure electronic cash based on blind multisignature schemes. In *Financial Cryptography and Data Security*, pages 236–250. Springer, 2006.
- [11] Minh Nguyen Hieu and Hung Dao Tuan. New multisignature schemes with distinguished signing authorities. In *Advanced Technologies for Communications (ATC), 2012 International Conference on*, pages 283–288. IEEE, 2012.
- [12] Debasish Jena, Sanjay Kumar Jena, and Banshidhar Majhi. A novel blind signature scheme based on nyberg-rueppel signature scheme and applying in off-line digital cash. In *Information Technology,(ICIT 2007). 10th International Conference on*, pages 19–22. IEEE, 2007.
- [13] Stefan Köpsell, Rolf Wendolsky, and Hannes Federrath. Revocable anonymity. In *Emerging Trends in Information and Communication Security*, pages 206–220. Springer, 2006.
- [14] Cheng-Chi Lee, Min-Shiang Hwang, and Wei-Pang Yang. A new blind signature based on the discrete logarithm problem for untraceability. *Applied Mathematics and Computation*, 164(3):837–841, 2005.
- [15] L Lopez-Garcia, L Martinez-Ramos, and F Rodriguez-Henriquez. A comparative performance analysis of several blind signature schemes. In *Electrical Engineering, Computing Science and Automatic Control, 2008. CCE 2008. 5th International Conference on*, pages 310–315. IEEE, 2008.
- [16] Kevin S McCurley. The discrete logarithm problem. In *Proc. of Symp. in Applied Math*, volume 42, pages 49–74, 1990.
- [17] Daniel Menasce. Scaling for e-business. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proceedings. 8th International Symposium on*, pages 511–513. IEEE, 2000.
- [18] Shirow Mitomi and Atsuko Miyaji. A general model of multisignature schemes with message flexibility, order flexibility, and order verifiability. *IEICE Transac-*

- tions on Fundamentals of Electronics, Communications and Computer Sciences*, 84(10):2488–2499, 2001.
- [19] Sujata Mohanty and Banshidhar Majhi. A secure multi authority electronic voting protocol based on blind signature. In *Advances in Computer Engineering (ACE), 2010 International Conference on*, pages 271–273. IEEE, 2010.
- [20] Kazuo Ohta and Tatsuaki Okamoto. A digital multisignature scheme based on the fiat-shamir scheme. In *Advances in Cryptology—ASIACRYPT’91*, pages 139–148. Springer, 1993.
- [21] Patiwat Panurach. Money in electronic commerce: Digital cash, electronic fund transfer, and ecash. *Communications of the ACM*, 39(6):45–50, 1996.
- [22] H Petersen, P Horster, and M Michels. Blind multisignature schemes and their relevance to electronic voting. In *11th Annual Computer Security Applications Conference, IEEE Press*, pages 149–155. Citeseer, 1995.
- [23] Weidong Qiu. Converting normal dlp-based signatures into blind. *Applied mathematics and computation*, 170(1):657–665, 2005.
- [24] Zulfikar Amin Ramzan. *Group blind digital signatures: Theory and applications*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [25] René Schoof. Four primality testing algorithms. *arXiv preprint arXiv:0801.3840*, 2008.
- [26] Victor RL Shen, Yu Fang Chung, Tzer Shyong Chen, Yu An Lin, et al. A blind signature based on discrete logarithm problem. *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL*, 7(9):5403–5416, 2011.
- [27] Nitu Singh and Sumanjit Das. Cryptanalysis of blind signature schemes. *IJCSNS*, 14(5):73, 2014.
- [28] Changji Wang and Hennong Xuan. A simpler restrictive partially blind signature. In *Pervasive Computing and Applications, 2006 1st International Symposium on*, pages 519–523. IEEE, 2006.

- [29] Stallings William and William Stallings. *Cryptography and Network Security*, 4/E. Pearson Education India, 2006.