Review on Multisignature Schemes Based Upon DLP

A thesis submitted in the partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Abhijit Nayak

(Roll: 110CS0137)

under the guidance of

Dr. Sujata Mohanty

NIT, Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, Orissa, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, Orissa, India.

May 12, 2014

Certificate

This is to certify that the thesis entitled "**Review on multisignature schemes based upon DLP**" submitted by **Abhijit Nayak** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering at the National Institute of Technology, Rourkela, is a record of his work carried out under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Dr. Sujata Mohanty
Dept. of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008

Acknowledgment

I would like to express my earnest gratitude to my project guide, Prof. Sujata Mohanty for giving me an opportunity to work under her guidance. This project could not have been possible without her inspiring guidance, valuable suggestions and constant support through the entire duration of my work.

I am indebted to all the other faculty members of Department of Computer Science and Engineering, NIT Rourkela for their valuable guidance and advices at appropriate times. I would like to thank all my friends for their help and assistance through this project.

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

Abhijit Nayak

110CS0137

Declaration of Authorship

I Abhijit Nayak, declare that this thesis titled, “A Review on Multisignature Schemes Based on DLP” and the work presented in it are my own. I confirm that:

- This work was done completely by me while in candidature for a B-Tech degree at this Institute.
- Where any portion of this thesis has previously been submitted for a degree or any other qualification at this Institute or any other University, this has been clearly stated in the references.
- Where I have consulted the published work of others, this is always clearly mentioned.
- Where I have quoted from the work of others, the source is also mentioned. This thesis is completely my own work with the exception of such quotations. If ever any dispute occurs, my supervisor is not responsible for that.
- I have acknowledged each and every main source of help.

Signed: _____

Date: _____

Abstract

In digital signature schemes a user is allowed to sign a document by using a public key infrastructure (PKI). For signing a document, the sender encrypts the hash of the document by using his private key. Then, the verifier uses the signer's public key to decrypt the received signature and to check if it matches the document hash. Generally a digital signature scheme demands only one signer to sign a message so that the validity of the signature can be checked later. But under some situations a group of signers is required to sign a message cooperatively, so that a single verifier or a group of verifiers can check the validity of the given signature. This scheme is known as a multisignature. A multisignature scheme is one of the tools in which plural entities can sign a document more efficiently than they realize it by trivially constructing single signatures. In general, in a multisignature scheme, the total signature size and the verification cost are smaller than those in the trivially constructed scheme. Thus, plural signers can collectively and efficiently sign an identical message. There are different base primitives describing the type of numerical problems upon which the underlying security scheme is based on. In this thesis, some of the most important DLP based multisignature schemes are presented. A categorization between these different existing schemes has been shown, along with their pros and cons.

Contents

List of Figures

Abbreviations

DLP: Discrete Logarithm Problem

IFP: Integer Factorization Problem

MS: Multisignature Scheme

OMS: Order-specified Multisignatrue Scheme

ECDLP: Elliptic Curve Discrete Logarithm Problem

CA: Certificate Authority

Chapter 1

Introduction

Being a part of the information age, we need to keep tabs on a variety of aspects of our life. With inflating volume of information, its value increases in manifolds day by day. Since the Internet serves as the quintessential mode of communication and a tool of commerce for tens of millions of people, security becomes a tremendously important issue to deal with. To be secured, information needs to fulfill three primary security goals named confidentiality – To be hidden from unauthorized accessed, integrity– To be protected from unauthorized changes and availability- To be available to an authorized entity when it is needed [1]. In order to ensure that the primary security goals are satisfied there are several security services and mechanisms to implement those services. In general security serves a variety of purposes, ranging from secure commerce and payments to private communications and protecting passwords. Cryptography is one such aspect of secure communication.

1.1 The Purpose of Cryptography

Cryptography, an ancient art, can be considered as the science of writing in secret code. Following the widespread development of computer communications new forms of cryptography came into existence. In case of data and telecommunications, cryptography is indispensable when communicating over any insecure medium, which includes pretty nearly any network, predominantly the Internet. Cryptography not only protects data from malice or modification, but also used for user authentication. The cryptographic schemes can be categorized into 3 categories to accomplish these goals: secret key cryptography, public-key cryptography, and hash functions [2].

1.2 Secret Key Cryptography

In secret or symmetric key cryptography the sender encrypts the message and sends it by a key say *k*. The receiver decrypts the message after receiving the message by using the same key *k*. The assumption is based upon the fact that here, both the sender and receiver use a common key and the transmission of the message and the key of cipher text is done in an insecure channel. This system is vulnerable and flawed if the key k is leaked and it is known to the adversary.

1.3 Public Key Cryptography

To overcome the problems of the symmetric key cryptography or the common key cryptography public key cryptosystem or public key encipherment is used. This scheme is similar to that of symmetric key cryptosystem, including few exceptions. Actually two keys are used instead of one, one public key and one private key. Before sending the message the sender encrypts it with the public key of the receiver. The receiver decrypts the message by using his own private key.

1.4 Digital Signature

A digital signature verifies the authenticity of an electronic document or digital message. The common use of digital signatures is to identify electronic entities for online transactions. A user is convinced to believe that the message was created by a known legitimate sender, such that later the sender cannot deny the fact that he had sent the message and that the message was not altered during transmission [3]. A digital signature uniquely identifies the originator of digitally signed data and also ensures the integrity of the signed data against tampering or corruption. Digital signatures are commonly used for the software distribution, authenticate online entity, and verify the origin of digital data. It also ensure the integrity of digital data against tampering, financial transactions, and in other case where it is important to detect forgery attack. In Figure 1.1 an entire digital signature procedure is shown [39].

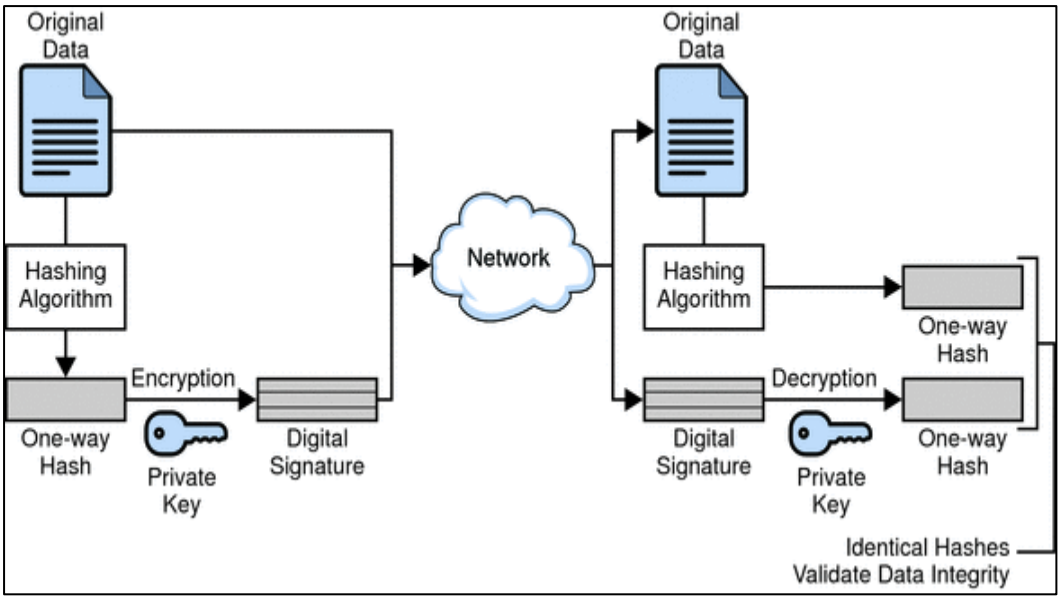

Figure 1.1: Digital signature procedure

1.4.1 Authentication

A message source is authenticated by digital signature. By the validation of the signature it is confirmed that the message was sent by that user, where user being the requester. Authenticity in digital signature means that the message or the user is valid [4,5].

1.4.2 Non-repudiation

Non-repudiation is a vital feature of digital signature. By virtue of this property, a signer simply cannot deny at a later point of time that he had not signed that [4, 5].

1.4.3 Integrity

The integrity of the message can be maintained even if we sign the whole message because the same signature cannot be obtained if the existing message is changed. With the help of hash functions, signing and verifying is done in case of digital signature so that the integrity of the message can be preserved [4, 5].

1.4.4 Attacks on Digital Signature

This section describes attack on digital signature. Key-Only attack, Known message attack and Chosen-Message attack are some attacks on DS. If the attack is successful, the result is a forgery. We can have two types of forgery [6, 7]. In a cryptographic digital signature system, digital signature forgery is the ability to create a pair consisting of a message and a signature that is valid for message, and message has not been signed by the legitimate signer [8]. Existential and Selective are the two types of forgery.

- *Existential Forgery*

In an existential forgery the attacker is able to create a valid signature-message pair, but the attacker cannot use this pair really. This type of forgery is probable, but the attacker cannot benefit from it [8].

- *Selective Forgery*

In the selective forgery, the attacker is able to forge signers signature on a message. The attacker gets benefit from this forge unlike existential forgery. The probability of such forge is low [8].

1.5 Multisignature

Being a society oriented signature scheme, a multisignature allows multiple signers to sign the same message cooperatively and in a simultaneous manner. A trifling solution says that every signer should sign the message using a normal signature scheme respectively. Clearly in this simple solution the security requirement of the multisignature scheme depends on the security of underlying signature schemes. It comes with the cost that both the data expansion and the computation costs for verification increases linearly as the number of signers grow in the group. Two

additional properties, submitted by Harn are to be satisfied in order to achieve an optimal multisignature scheme [9]:

1) The size of a multisignature should be identical to that of an individual signature.
2) The verification process of a multisignature should be almost identical to that of an individual signature.

Therefore, in an ideal multisignature scheme, the size of signatures as well as the computation costs for verification should be independent of the number of signers participating in signing. A multisignature procedure is shown in Figure 1.2

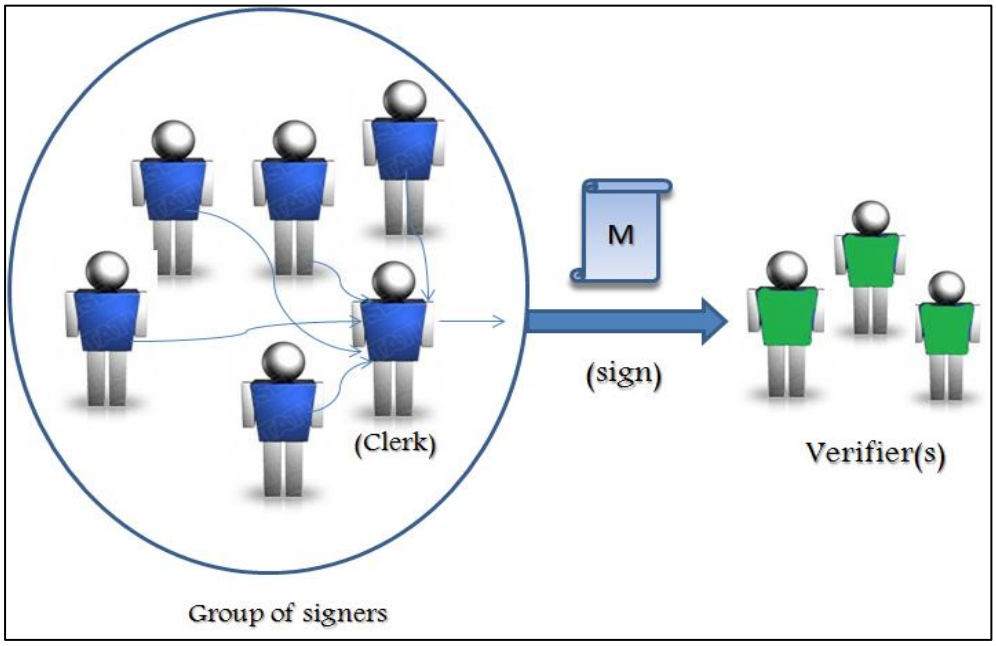

Fig 1.2: Multisignature procedure

1.6 Evolution of Different Multisignature Schemes

In 1983, Itakura and Nakamura proposed the first multisignature scheme, which was a modification of the RSA cryptosystem such that the module taken was the product of three primes rather than just two [10]. Since then various other multisignature schemes have been proposed based on different base primitives. Base primitive denotes the kinds of numerical problems on which the security of the multisignature schemes depends. Generally base primitives include IFP, DLP and ECDLP.

After that many multisignature schemes had been proposed in [11-16] but the first multisignature, based on DLP was proposed by Harn in 1994 [17]. As here we focus only on schemes based upon DLP we will discuss [11-13] and many more under different categories.

1.7 Types of Multisignature

Generally digital multisignature schemes can be classified into two classes depending on the authority of the signers.

1) Multisignatures with undistinguished signing authorities.
2) Multisignatures with distinguished signing authorities.

For the first class of multisignature scheme, all the members present in the signing group bear the same signing authorities for the entire message where as in the second class of multisignature scheme every member gets his own distinguished signing authority [11].

All the digital signature schemes can be further classified into two classes depending on the process of verification.

1) Multisignatures for specified group of verifiers.
2) Multisignatures for any verifier(s).

In the first case a group of signers cooperatively sign a message for a specified group of verifiers, and the validity of the multisignature can be checked only by all the verifiers in that specified group together, but in second case any number of verifier can validate the signature for the message [12].

1.8 Signature Structure

A multisignature scheme is considered to be structured only if the group of signers is structured. The signing order of the entities plays a vital role. Actually it signifies a special meaning when signers sign a document sequentially. Such a signature structure is called serial. Alternatively, if partial signatures are created by all or part of the

signers in an arbitrary order and combined to create a complete signature, the signing order of the signers has no meaning. Such a signature structure is called parallel [13].

1.9 Organization of Thesis

The organization of the thesis is as follows: Chapter 2 discusses some mathematical preliminaries; In Chapter 3 the multisignature schemes for distinguished signing authorities has been discussed. Chapter 4 describes structured multisignature schemes. Chapter 5 includes description of multisignature schemes foe specified group of verifiers. Finally, we conclude with Chapter 6 by giving few observations.

Chapter 2

Mathematical Preliminaries

Following basic notations, definitions and models are used throughout this thesis.

2.1 Notation and Terminology

All groups discussed in this thesis are assumed to be abelian. Groups of prime order have useful properties and are widely used in cryptography. All groups of prime order are cyclic.

A group G is said to be cyclic if there is an element $g \in G$, such that for each $g' \in G$, there is an integer a with $g' = g^a$. Such an element is called a generator of G [18]. For any prime integer p, the field of integers modulo p is denoted by Z_p. The cyclic multiplicative group of nonzero elements in Z_p is denoted as Z_{p^*}.

2.2 Discrete Logarithmic Problem (DLP)

Particularly in abstract algebra and its applications, discrete logarithms are group theoretic analogues of ordinary logarithms. An ordinary logarithm log(a, b) gives a solution for the equation $a^x = b$ over the real or complex numbers. Likewise, if g and h are elements of a finite cyclic group G then a solution x of the equation is called a discrete logarithm to the base g of h in the group G. Briefly, if G is a finite group, the problem discrete logarithm in G is the following computational problem: given elements α and β in G, determine an integer x such that, $\alpha^x = \beta$, provided that such an integer exists [8].

2.3 Computational Diffie-Hellman problem

The Diffie-Hellman problem can be described as follows. Let G be a cyclic group of order q. If g is a generator of some group, preferably the multiplicative group of a finite

field, and x, y are randomly chosen integers then CDH assumption states that, given(g, g^a, g^b) for any randomly choosen generator g and $a, b \in \{0, \dots, q-1\}$ it is computationally infeasible to compute the value of [19].

2.4 The Integer Factorization Problem

There are many fast algorithms for multiplying two given large prime numbers. On the other hand, it is considerably difficult to find the prime factors if the product of two large primes is given. The perceptible difficulty of factoring large integers forms the foundation of some modern cryptographic algorithms. Many schemes rely on the difficulty of factoring large integers, such as the RSA encryption algorithm and the Blum Blum Shub cryptographic pseudorandom number generator [20, 21]. If quick factorization of large primes is possible, these algorithms would not be secure anymore.

2.5 Safe Primes

Their relationship with the strong primes is what makes them safe prime. By definition a prime q is said to be a strong prime if $q+1$ and $q-1$ both have large prime factors. For a safe prime, $q = 2p + 1$, the integer p is a large prime factor. The importance of safe primes is realized when they are used in discrete logarithm-based techniques like Diffie-Hellman key exchange If $2p + 1$ is a safe prime, the multiplicative subgroup of numbers modulo. $2p + 1$ has a subgroup of large prime order. Safe primes are used to minimize the modulus [22].

Chapter 3

Multisignature Scheme with Distinguished Signing Authorities

3.1 Introduction

In general, all the constituent members of the group in a multisignature scheme are endowed with the same signing rights for the entire document, but under certain situations each member needs to have his own distinguished signing authority. As an example, say a governing authority receives the annual performance report about an organization. It is the partial contents from different departments upon which the report is based on. Distinguished responsibility makes each department to authorize its partial contents. The readers are allowed to explore some authorized partial contents selectively because of the abject requirement of confidentiality. Simultaneously, the validation of the relationship between an entire report and its partial contents is done by them. The accuracy of the partial contents is verified, too. So the two additional properties [9] that must be satisfied for any MS with distinguished signing authorities are

I. Without revealing the entire message, partial contents can be easily verified.
II. There should be distinguished signing authority for each member

In this work, we present several of the most relevant multisignature schemes with distinguished signing authorities which are based upon DLP.

3.2 Review of Harn's Scheme

In 1999, L. Harn first proposed an MS with distinguished signing authorities which is used in several cryptographic applications [9]. As an example, a credit card company or a telephone company or a medical insurance company can set up a joint

scheme to issue smart cards for customers. This scheme is based on DLP and each member in the signing group is allowed to access only partial contents of the entire document. Then each member is given distinguished singing authorities for his partial contents. The scheme in general consists of three phases: the system initiation phase, the individual signature generation and verification phase, and the multisignature generation and verification phase.

(i) System initialization phase

Generally three system parameters p, g and h, where p is some large prime number, g is a primitive element in $GF(p)$ and h being the one-way hash function are used. Let the signing group is $\{U_1, U_2, \dots U_n\}$. Each signer randomly picks an integer x_i from $[1, p-1]$ as his private key and computes his public key $y_i = g^{x_i} \, mod \, p$ for $i = 1, 2, 3 \dots, n$. Then the group public key $y = \prod_{i=1}^{n} y_i \, mod \, p$ is calculated.

(ii) Individual signature generation and verification phase:

Suppose there are n signers $U_1, U_2, \dots U_n$ who want to sign the distinguished messages $m_1, m_2, \dots, m_n$ respectively.

(a) Individual signature generation:

Every signer U_i randomly picks a number k_i from $[1, p-1]$ and computes $r_i = g^{ki} mod \, p$, then broadcasts r to all signers. Also U_i broadcasts $h(m_i)$ to all signers. Once all the r_i's for $i = 1, 2, 3 \dots, n$ are available by means of the broadcast channel, each signer calculates the value $r = \prod_{i=1}^{n} r_i \, mod \, p$. Then U_i tries to find the solution for individual signature equation

$$s_i = x_i m' - k_i r \, mod \, (p-1)$$

To obtain the value of s_i where $m' = h\big(h(m_1), h(m_2), \dots, h(m_n)\big)$ and $h\big(m_j\big)$can be received from the broadcast which is yielded by other signers U_j $(j = 1, 2, 3 \dots, n)$ and j ≠ i. Therefore the set (r_i, s_i) is considered as the individual signature for the message signed by the signer U_i.

(b) Individual signature verification:

Once the individual signature (r_i, s_i) is received by the clerk from each U_i, he verifies the validity of the signature by checking verification equation

$$y_i^{m'} = r_i^{r}\, g^{s_i} \bmod p$$

It is considered that the individual signature (r_i, s_i) received from each U_i is verified if the above equation holds.

(*ii*) *Group signature generation and verification phase:*

(a) Group signature generation:

After receiving all the individual signatures the clerk verifies them and computes

$$r = \prod_{i=1}^{n} r_i \bmod p$$

$$s = \sum_{i=1}^{n} s_i \bmod p$$

Where (r, s) is considered as the multisignature signed by all signers $U_1, U_2, \ldots U_n$ on the message $\{ m_1, m_2, \ldots, m_n\}$

(b) *Group signature verification:*

The group signature is verified by

$$y^{m'} = r^r g^s \bmod p$$

Where

$$y = \prod_{i=1}^{n} y_i \bmod p$$

and

$$m' = h\big(h(m_1), h(m_2), \ldots, h(m_n)\big)$$

Advantages

a) Each signer gets distinguished signing authority because each of them is responsible for preparing a section of message.
b) In place of signing $m' = h(m_1, m_2)$, each signer needs to sign $m' = h(h(m_1), h(m_2))$. The computation of $h(h(m_1), h(m_2))$ is faster than that of $h(m_1, m_2)$ because of the fact that each signer needs only to compute his own $h(m_i)$ and the other $h(m_i)$ has been computed by the other signer.
c) There are certain situations where some of the verifiers are only allowed to access partial contents of the message, the verification of these partial contents can still be done using the group public key without revealing the whole message. This feature is achieved by providing just the one − way hash values of the inaccessible contents to the verifier. As an example, by disclosing m_1, and $h(m_2)$ to the verifier, the verifier can still confirm the authenticity of m_1.

Disadvantages

a) Later Li et al. showed that Harn's scheme is vulnerable to their attack [23]. According to Li et al. a valid multisignature for any message can be forged by a malicious insider attacker without any private keys of the other signers. It is almost impossible to detect the insider attack for any outsider or verifier. The CA needs each user to show that he actually knows the secret exponent of his public key in order to prevent their attack. Therefore this attack points some weakness in Harn's scheme by increasing load and causing inconvenience for CA and users.
b) Additionally, in Harn′s scheme, it was not possible for an individual signer to prove his own distinguished signing authority although the fact he actually signed only for his partial content is true. There is no evidence to distinguish the signing authorities. The cause being so obvious that in Harn's scheme, all individual signatures and multisignatures are generated on the same hash digest of the hash digests of all the partial contents. Hence, the use of the individual signatures as proof for the partial content is unacceptable. A new multisignature

scheme with distinguished signing authorities is proposed in order to guard against Li et al.'s attack without the help of CA [11].

3.3 Review of Hwang's Scheme

In 2003 S. Hwang et. al. proposed a new multisignature scheme with distinguished signing authorities which is secured against Li et al.'s attack without the help of CA [11]. With the help of individual evidence provided by this new scheme puzzlement over authority due to malice can be avoided. The entire scheme consists of 3 phases.

Parameters for System and Signing Groups

Let p and q are two publicly known large prime numbers such that q|p-1. The integer g is considered as a public generator with order q in $GF(p)$, and the function $h()$ being a public one-way hash function. Let the signing group is $\{U_1, U_2, \dots, U_n\}$. Every member U_i randomly picks his private key $x_i \in Z_{q^*}$ and calculates his public key $y_i = g^{x_i} \bmod p$. Then the group public key

$$Y = \prod_{i=1}^{n} (y_i)^{y_i} \bmod p$$

is calculated.

Multisignature Generation Phase

Let the signing group $\{U_1, U_2, \dots, U_n\}$ wants to obtain the multisignature for the message $M = m_1 \| m_2 \| \dots \| m_n$. The member U_i is only in charge for the partial content m_i, for $i = 1, 2, 3 \dots, n$.

Step1: Each member U_i picks a random integer $k_i \in Z_{Q^*}$ and computes $r_i = g^{k_i} \bmod p$ and $h(m_i)$ for $i = 1, 2, 3 \dots, n$. Then each member U_i broadcasts r_i and $h(m_i)$ to the other $n - 1$ members and a predetermined clerk C.

Step2: The commitment value r is calculated by each member U_i

$$r = \prod_{i=1}^{n} r_i^{h(h(m_i),r_i)} \ mod\ p$$

The clerk also calculates the commitment value r.

Step3: Each member U_i finds the solution s_i satisfying the following condition.

$s_i = x_i y_i H + r k_i\ h(h(m_i), r_i)\ (mod\ q)$, with $H = h\big(h(m_1), h(m_2), \dots, h(m_n)\big)$.

Then each member U_i transmits his individual signature (r_i, s_i) to the clerk.

Step4: The clerk verifies each the individual signature $(r_i,\ s_i)$ by means of the equation $g^{s_i} \equiv (y_i)^{y_i H} \times r_i^{h(h(m_i),r_i)}\ (mod\ p)$ after receiving all of the individual signatures (r_i, s_i)'s. If all of the individual signatures are legal, then the clerk generates the multisignature (r,s) by computing

$$s = \sum_{i=1}^{n} s_i\ mod\ q$$

Finally, (r, s) is the multisignature for the message $M = m_1 \| m_2 \| \dots \| m_n$.

Multisignature Verification Phase

The multisignature $(r,\ s)$ is verified by means of the equation $g^s \equiv Y^H \times r^r\ (mod\ p)$.Why the equation $g^s \equiv Y^H \times r^r\ (mod\ p)$ can be used to verify the multisignature (r, s) is shown in the following:

$$\begin{aligned} g^s &\equiv g^{\sum_{i=1}^{n} s_i} \\ &\equiv g^{\sum_{i=1}^{n} \left(x_i y_i h\left(h(m_1),h(m_2),\dots,h(m_n)\right) + r k_i\ h(h(m_i),r_i)\right)} \\ &\equiv g^{\sum_{i=1}^{n} x_i y_i h\left(h(m_1),h(m_2),\dots,h(m_n)\right)} \times \left(g^{\sum_{i=1}^{n} k_i\ h(h(m_i),r_i)}\right)^r \\ &\equiv Y^H \times r^r\ (mod\ p) \end{aligned}$$

The partial contents of the message $m_1 \| m_2 \| \dots \| m_n$ can be verified without disclosing the entire document. If the verifier is only allowed to read the partial content m_i, then he will receive $h(m_1) \| h(m_2) \| \dots \| h(m_{i-1}) \| m_i \| h(m_{i+1}) \| \dots \| h(m_n)$ to verify the multisignature (r, s)

This new scheme provides additional evidence which can be used by the members in order to prove their distinguished signing authority. This feature was not there in Harn's scheme. Moreover in this scheme, an individual signature (r_i, s_i) as well as the multisignature (r, s) can show the relationship between an entire document, its partial content, and the signing members. So that each member can show that he only has signing responsibility for the partial content for which he has signed previously.

However Hieu showed that the computation and communication costs for generation of the multisignature will be significantly affected by the number of signers in the group [24]. As the no of signers increases in a group, the time taken for the generation of the multisignatures also increases significantly.

3.4 Review of Hieu's Scheme

In 2012 hieu et al. proposed two multisignature schemes with distinguished signing authorities [24]. The underlying mathematical problems for the two signatures schemes include solving discrete logarithmic problem and finding roots modulo prime. Each of the two proposed schemes consists of three phases named key generation, multisignature generation and multisignature verification respectively.

First Scheme

Let the signing group $\{U_1, U_2, \dots, U_n\}$ wants to produce a multisignature for the message $M = m_1 \| m_2 \| \dots . \| m_n$. The responsibility of the member U_i is only for the partial content, say $m_i (i = 1, 2, 3 \dots, n)$.

The Key Generation Phase

Assuming a group of n signers and a trusted clerk, the following parameters are defined:

Step 1: A trusted clerk chooses a large prime p, a prime divisor q correspondingly with $q \mid (p - 1)$, and a one-way hash function such as $SHA - 1(H = h(M))$

Step 2: $x_1, x_2, \dots, x_n$:group members′ secret keys such that $1 < x_i < q$, x_i is selected randomly and known only by the member U_i.

Step 3: $y_1, y_2, \dots, y_n$: group members' public keys such that $y_i = \alpha^{x_i} mod\ p$ is computed and published by the group members U_i (α is generator of the cyclic group of order $q \in Z_{p^*}$). Adding or deleting a member i requires adding or deleting the corresponding y_i by the clerk.

Step 4: The clerk computes group public key Y for all signers,

where

$$Y = \prod_{i=1}^{n} (y_i)^{y_i} \ mod\ p$$

The Multisignature Generation Phase

The scheme requires the clerk and other signing group members to carry out an exchange of data during the multisignature generation process.

Step1: Each signer selects random number $k_i \in Z_{q^*}$ and computes $r_i = \alpha^k mod\ p$. Then each signer U_i sends $r_i$to the clerk.

Step2: The clerk computes the common randomization value

$$R = \prod_{i=1}^{n} r_i^{h(m_i)} \ mod\ p$$

and computes the values $E = h(R \| M)$ and $H = h(h(m_1) \| h(m_2) \| \dots . \| h(m_n))$. Then he sends (E, H) to each of the signers.

Step 3: Each signer computes its signature share s_i as follows

$s_i = k_i h(m_i) H + x_i y_i E \ (mod\ q)$, then each signer U_i sends s_i to the clerk.

Step 4: Once the clerk receives the individual signature $(r_i,\ s_i)$ from i signers, he needs to verify the validity of this individual signature. The clerk checks the signature of the individual as follows

$$\alpha^{s_i} \equiv {y_i}^{y_i E} {r_i}^{h(m_i)H} \text{ mod p}$$

If all of the individual signatures are legal, then the clerk generates the multisignature (R, S) by computing

$$S = \sum_{i=1}^{n} s_i \ mod\ q$$

Finally, (R, S) is the multisignature for the message $M = m_1 \| m_2 \| \ldots . \| m_n$

The Multisignature Verification Phase

Prior to verifying the signature of a signed message, the parameters (p, α, Y) are made available to the verifier in an authenticated manner.

Verification of the multisignature is performed using the group public key Y.

Step 1: Using the multisignature (R, S) to compute $\alpha^{S'} \equiv Y^E R^H \ mod\ p$

Step 2: Compare values S' and S. If $S' = S$, then the signature is valid. Otherwise the signature is false.

The partial contents of the message $m_1 \| m_2 \| \ldots . \| m_n$ can be verified without disclosing the entire document. If the verifier is only allowed to read the partial content m_i then he will receive $h(m_1) \| h(m_2) \| \ldots . \| h(m_{i-1}) \| m_i \| h(m_{i+1}) \| \ldots . \| h(m_n)$ to verify the multisignature (R, S)

Second Scheme

Let the signing group $\{U_1, U_2, \ldots, U_n\}$ wants to generate the multisignature for the message $M = m_1 \| m_2 \| \ldots . \| m_n$. The member U_i is only responsible for the partial content, m_i for $(i = 1, 2, 3 \ldots, n)$

The Key Generation Phase

This scheme uses the prime modulus having the structure $p = Nq^2 + 1$, where q is a large prime number ($|q| \geq 160$) and N is such even integer that ($|p| \geq 1024$) bits. Assuming a group of n signers and a trusted clerk, the following parameters are defined:

Step 1: A trusted clerk generates a large prime p, a prime divisor q having the structure $p = Nq^2 + 1$ correspondingly with $q^2 | p - 1$ and a one-way hash function such as $SHA - 1$ ($H = h(M)$).

Step 2: $x_1, x_2, \dots, x_n$:group members' secret keys such that $1 < x_i < q$, x_i is selected randomly and known only by the member U_i.

Step 3: $y_1, y_2, \dots, y_n$: group members' public keys such that $y_i = x_i{}^q mod\ p$ is computed and published by the group members U_i. Adding or deleting a member i requires adding or deleting the corresponding y_i by the clerk.

Step 4: The clerk computes group public key Y for all signers:

$$Y = \prod_{i=1}^{n} (y_i)^{y_i} \ mod\ p$$

The Multisignature Generation Phase

The scheme requires the clerk and other signing group members to carry out an exchange of data during the multisignature generation process.

Step 1: Each signer selects random number $k_i \in Z_{q^*}$ and computes $r_i = k_i{}^q mod\ p$. Then each signer U_i sends $r_i$to the clerk.

Step2: The clerk computes the common randomization value

$$R = \prod_{i=1}^{n} r_i{}^{h(m_i)} \ mod\ p$$

and computes the values $E = h(R\|M)$ and $H = h(h(m_1)\|h(m_2)\| \ldots . \|h(m_n))$. Then he sends (E, H) to each of the signers.

Step 3: Each signer computes its signature share s_i as follows

$$s_i{}^q = k_i h(m_i)H + x_i y_i E \ (mod\ q)$$

Then each signer U_i sends s_i to the clerk.

Step 4: Once the clerk receives the individual signature $(r_i,\ s_i)$ from i signers, he needs to verify the validity of this individual signature. The clerk checks the signature of the individual as follows $s_i{}^q \equiv y_i{}^{y_i E} r_i{}^{h(m_i)H}$ mod p. If all of the individual signatures are legal, then the clerk generates the multisignature (R, S) by computing

$$S = \sum_{i=1}^{n} s_i \ mod\ q$$

Finally, (R, S) is the multisignature for the message $M = m_1\|m_2\| \ldots . \|m_n$

The Multisignature Verification Phase

Prior to verifying the signature of a signed message, the parameters (p, q, Y) are made available to the verifier in an authenticated manner. Verification of the multisignature is performed using the group public key Y

Step 1: Using the multisignature (R, S) to compute $S'^q \equiv Y^E R^H \ mod\ p$

Step 2: Compare values S' and S. If $S' = S$, then the signature is valid. Otherwise the signature is false.

The partial contents of the message $m_1\|m_2\| \ldots . \|m_n$ can be verified without revealing the whole document. If the verifier is only allowed to read the partial content m_i then he will receive $h(m_1)\|h(m_2)\| \ldots . \|h(m_{i-1})\|m_i\|h(m_{i+1})\| \ldots . \|h(m_n)$ to verify the multisignature (R, S).

Hieu and Hung's schemes provide individual evidence to prevent confusion over authority due to malice [24]. Moreover, the new schemes provide generation of

the multisignature possessing with internal integrity. Compared to Hwang et al.'s scheme, the proposed schemes have the advantages of reducing computation and communication costs and making the size of the multisignature flexible.

Chapter 4

Structured Multisignature Scheme

4.1 Introduction

When a document is signed by multiple entities, it is the signing order that reflects the part of each signer and signatures having different signing orders are regarded as multisignatures with different meanings. Consider a company as an example. Generally, a document is signed by the head of a section only after other members of the section have signed it. Some other examples are banks and command structures. Certainly the signing order has a little relevance to authentication. Nevertheless there are other aspects that should be taken into account, such as the legal responsibility of the signers. This could affect their ranking and can be determined by the signing order. As per the applications, different signing orders may be required. An MS, in which the set of the signers as well as the signing order can be verified, is called an OMS [24].

Additionally, each signer may wish to sign only after the previous signers have signed the message, and the verifier may need to check that the correct order has been followed. Two types of signing order may be considered: 1) serial signing, which allows the verifier to detect the signing order from the signature, 2) parallel signing, there is no way, a verifier can detect the signing order from a signature [13] . In [13, 25, 26, 27, 28], we can see several OMS but in this thesis, we focus on OMS having the discrete logarithm problem as base primitive, therefore we will review the schemes [13,28].

4.2 Review of Burmester's Scheme

System Initialization

According to Burmester's scheme two large primes p and q are chosen such that $q \mid p-1$ [13]. A primitive-root g is selected over the cyclic group $GF(p)$ and $h()$ denotes a one-way collision resistant cryptographic hash function. Supposed $(U_1, U_2, \dots, U_{n-1}, U_n)$ are the signature order. All the signers choose an integer as their private keys $x_i \in Z_{q^*}$ at random, compute their public keys sequentially as follows: $y_1 = g^{x_1} mod\ p, \quad y_i = (g.y_{i-1})^{x_i} mod\ p$ .The public key of the group $(U_1, U_2, \dots, U_{n-1}, U_n)$ is $y = y_n(\ mod\ p)$.

Structured Multi − signature Generation

Signature parameter R generation phase:

(1) The first signer U_i randomly chooses an integer $k_i \in Z_{q^*}$ and computes

$$r_1 = g^{k_1} mod\ p$$

If gcd(r_1 , q) ≠ 1 then chooses new k_1 again.

(2) For $i \in \{2, 3 \dots, n\}$ U_{i-1} gives r_{i-1} to U_i

U_i randomly chooses $k_i \in Z_{q^*}$ and computes $r_i = (r_{i-i})^{x_i} g^{k_i}\ mod\ p$

If gcd(r_i , q) ≠ 1 then U_i chooses k_i again until gcd(r_i , q) =1

(3) $R = r_n$.

Signature parameter S generation phase:

$U_1, U_2, \dots, U_{n-1}, U_n$ generate S together as follows.

(1) U_1 computes $s_1 = x_1 + k_1 Rh(R, M) mod\ q$

(2) For $i \in \{2, 3 \dots, n\}$ U_{i-1} gives s_{i-1} to U_i

U_i verifies that $g^{s_{i-1}} \equiv y_{i-1} {r_{i-1}}^{R.hash(R,M)} (mod\ p)$ and if so computes

$$s_i = (s_{i-1} + 1)x_i \ + k_i Rh(R, M) mod\ q$$

(3) $S = \ s_n$

Multisignature. (R, S) is the multisignature on M by $(U_1, U_2, \dots, U_{n-1}, U_n)$.

Verifier U_v verifies the multisignature (R, S) by checking the congruence

$$g^S \equiv y.R^{R.h(R,M)} mod\ p$$

Zhang showed that Burmester's scheme is not safe. An attacker can forge certain messages by forging his own public key, signature parameter and signature, provided he is the signer else he can do the forgery by forge signature parameters [29]. Moreover zhang proposed four inside attack methods to the Burmester's structured signature scheme by which the attacker can replace $\{U_{t+1}, U_{t+2}, \dots, U_i\}(t \in [1, i-1])$ to sign the message without authority. Later he proposed an improved scheme.

4.3 Review of W. Luo's Scheme

This scheme adds a signature verifier to check the signers' signature parameters in signature process, which can resist forgery attack [28]. When the signers finish generating their signature parameters, then they generate a parameter according to their signature parameters, after that they send the parameter to the signature verifier to check whether their signature parameters is valid. In this way, the improved scheme can resist forgery attack. This scheme is divided into three phases: system initialization, signature process and signature verification.

System Initialization

(1)The system chooses two large primes p and q such that $q \mid p-1$, let g be the primitive-root of the cyclic group $GF(p)$, $h()$ denotes a one-way collision resistant cryptographic hash function. are the signature order.

(2) The signers $(U_1, U_2, \dots, U_{n-1}, U_n)$ respectively choose $(x_1, x_2, \dots, x_n) \in Z_{q^*}$ randomly as their private keys, and compute their public keys sequentially as follow $y_1 = g^{x_1} \, mod \, p$, $y_i = y_{i-1} g^{x_1} mod \, p$

(3) Signers open system parameters and their public keys, meanwhile, preserve their private keys.

Signature Process

Signature Parameters R Generation

The signature verifier U_v randomly chooses an integer $k_v \in Z_{q^*}$ and computes his signature parameters $r_v = g^{k_v} mod \, p$, then publish r_v to all the signers. For $(i = 1, 2, 3 \dots, n-1)$, U_i randomly selects their own integer $k_i \in Z_{q^*}$, and computes their signature parameters sequentially as follows: $r_1 = g^{k_1} \, mod \, p$, $r_i = r_{i-1} g^{x_1} \, mod \, p$. Then broadcast to all of the users (signers and verifier). The last signer U_n computes the signature parameters $R = r_n$.

When the signer U_i completes generating his signature parametersr_i, then he computes $w_i = {r_v}^{k_i} \, mod \, p$ and sends w_i to the system verifier U_v to verify whether his signature parameters is forgery.

Signer's Signature Parameters Verification

In this process, the system verifier U_v verifies all of the signers' signature parameters are forgery or not. If someone's signature parameters is fake, then signature system is unsafe. Because the inside and outside attacker can forge an unauthorized message by forge his signature parameters. So, all signers' signature parameters through verifying, this scheme can resist forgery attack. The method as follow:
When system verifier U_v received the signer U_i's w_i, he verifies

$$w_i \equiv (r_i . {r_{i-1}}^{-1})^{k_v} \, mod \, p$$

$${r_i}^{k_v} \, mod \, p \neq 1 \, mod \, p$$

If the one of two equations are not verify, it is means that the signer U_i's signature parameters is forgery, so the signer verifier must suspend the next signer U_{i+1} to generate his signature parameters and let U_i generate his signature parameters again until the signer U_i's signature parameters is valid. If all of the signers' signature parameters are valid, the process of signature parameters verification completed.

Generation of signature

The signer is U_i firstly computes the signature $s_1 = x_1 + k_1 Rh(R, M) mod\ q$ then sends (s_1, M) to the next signer. The signer U_i $(i = 1, 2, 3 \ldots, n)$, takes the following steps:

(1) Verifying $g^{s_{i-1}} \equiv y_{i-1} {r_{i-1}}^{Rh(R,M)} mod\ p$, if the equations establish, it means that the structured multisignature is valid, or reject the signature and judge the signature invalid.

(2) Firstly he computes the signature $s_i = s_{i-1} + x_i + k_i Rh(R, M) mod\ q$, then sends (s_i, M) to the next signer.

When all of the signers have finished signing the message M, the last signer sends (s_n, M) to system verifier U_v. Firstly, U_v computes $= r_n$, then verifies the equations that $g^{s_n} \equiv y_n . R^{R.h(R,M)} mod\ p$ establish or not. If the equations establish, it is means that the signature is valid, else judge the signature invalid, and terminate the signature.

Lou *et. al.* points out a structured multisignature scheme against forgery attack and shows that the new scheme can resist inside and outside forgery attack by verifying signature parameters [28].

Chapter 5

Multisignature Scheme for Specified Group of Verifiers

5.1 Introduction

Usually a single signer is sufficient to sign a message and the validity of the signature can be checked by any number of verifier(s). Nevertheless, there are situations in which a group of signers sign a message cooperatively and only all verifier in the specified group of verifiers together will be able to verify the validity of the multisignatures [12]. Often there are situations in which a single verifier cannot be trusted with the signature process. For example a board needs to approve a confidential document. It would be more reliable to handle the responsibility on a group of board members rather than trusting a single member with verifying the document and the signature.

5.2 Review of Laih and Yens' scheme

In 1996, Laih and Yen (LY) first proposed the concept of the multisignature scheme for a specified group of verifiers and it was based upon DLP [30]. In this scheme the group of signers, not only use each signer's private key but also the group public key of verifiers to sign a message. In both the multisignature generation phase and the multisignature verification phase, the group of signers and the group of verifiers need a clerk to assist them in signing messages and verifying multisignatures, respectively. Like all other multisignature schemes LY scheme consists of 3 phases namely, key generation, multisignature generation and multisignature verification.

Key generation phase:

Let $G_s = (u_{s_1}, u_{s_2}, \ldots, u_{s_n})$ be the group of n signers and $G_v = (u_{v_1}, u_{v_2}, \ldots, u_{v_m})$ be the group of m verifiers. In each group, there is a specified user, called clerk. The clerk u_{s_c} of the signers' group is responsible for verifying all partial signature signed by signer in G_s and combining them into multisignature. The Clerk u_{v_c} of the verifiers' group is responsible for assisting all verifiers in G_v to verify the multisignature.

The trusted center plays here a crucial role in performing this multisignature scheme. The trusted center selects 2 large primes 'p' & 'q' such that $q \mid p-1$. It also selects an element $g \in Z_{p^*}$ with order 'q'. Each $U_{s_i} \in G_s$ selects his private key $s_i \in Z_q$ and computes his public key $Y_{S_i} = g^{-s_i} \bmod p$. Each $U_{v_j} \in G_v$ selects his private key $v_j \in Z_q$ and computes his public key $Y_{V_j} = g^{-v_j} \bmod p$.Then G_S, and G_V respectively publish thir group public keys $Y_S{}'$ and $Y_v{}'$ where

$$Y_S{}' = \prod_{i=1}^{n} g^{-s_i} \bmod p$$

and $$Y_v{}' = \prod_{i=1}^{n} g^{-v_j} \bmod p$$

Multisignature generation phase:

All signers in G_s perform the following steps to generate the multisignature of a message M for the specified group G_v of verifiers:

1. Each $U_{s_i} \in G_s$ chooses a random integer r_i, computes $x_i = \left(Y_V{}'\right)^{r_i} \bmod p$ and and sends x_i to u_{s_c} .
2. U_{s_c}computes $x = \prod_{i=1}^{n} x_i \bmod p$ and broadcastes x to all signers in G_s .
3. Each $U_{s_i} \in G_s$ computes $e = h(x, M)$ and $w_i = r_i + e \,.\, s_i \bmod q$, then sends w_i to U_{s_c} .
4. Upon receiving all w_i $(i = 1,2, \ldots, n)$, u_{s_c} computes

$$e = h(x, M)$$

and $$w = \sum_{i=1}^{n} w_i \bmod q,$$

and sends M and its multisignature (e, w) to G_v.

Multisignature verification phase:

All verifiers in G_V perform the following steps to verify the multisignature of a message M :

1. Each $U_{V_j} \in G_V$ computes

$$X_j = \left(g^w . \left(Y_S{}'\right)^e\right)^{-v_j} \ mod\ p$$

and sends X_j to U_{V_c}.

2. U_{V_c} computes

$$X = \prod_{j=1}^{m} X_j \ mod\ \ p$$

and then broadcastes X to all verifiers G_V.

3. Each $U_{V_j} \in G_V$ verifies the validity of the multisignature (e, w) by checking if $e = h(x, M)$. If the equality holds, the multisignature (e, w) of M is indeed signed by G_S.

Yen's scheme showed that the clerk of a specified group of verifiers can alone verify the validity of multisignatures without the help of other verifiers . According to Xie, if the specified group of verifiers has ever verified the multisignature signed by the group of signers G_S and has new participant, they can cooperate to forge any message by the group public key adjustment because of the renewed members. It also states that if a specified group of verifiers has ever verified the multisignature signed by Gs, they can cooperate to forge the signature for any message by the secret key substitution due to the leaked secret key [32].

Moreover later Yen himself showed the vulnerability of the LY scheme to a new attack, called spoofing between a rebel of the verifiers and a cheating signer(s), in a multi-verifier, signature scheme with verifier specification [33].

5.3 Review of Hwang's Scheme

Later, Hwang et al. (HCC) proposed another multisignature scheme for a specified group of verifiers that can provide authenticity as well as confidentiality [34]. It is actually different from the previous schemes and the LY scheme. In this scheme, only all verifiers in the specified group together are able to recover the message from the multisignature and check if the message is signed by the group of signers. The group of signers and the group of verifiers also need a clerk, respectively. Brief review of the HCC scheme is described as follows.

Key Generation Phase:

Let G_S, G_V, U_{S_c} *and* U_{V_c} be the group of n signers, the group of m verifiers, the clerk of G_S and the clerk of G_V respectively. First of all, the trusted center also selects the same p, q, and g as those in the LY scheme. All signers in G_s share a common secrete key $S \in Z_{q^*}$. Each $U_{S_i} \in G_s$ selects his private key $s_i \in Z_{q^*}$ and computes his public keys $Y_{S_i}{}' = g^{s_i} mod\ p$ and $Y_{S_i}{}'' = g^{S.s_i} mod\ p$. Then, G_S publishes three group public keys Y_S, $Y_S{}'$, $Y_S{}''$

Where

$$Y_S = g^S\ mod\ p,$$

$$Y_S{}' = \prod_{i=1}^{n} Y_{s_i}{}'\ mod\ p$$

$$Y_S{}'' = \prod_{i=1}^{n} Y_{s_i}{}''\ mod\ p$$

All signers in G_V share common secret key $\in Z_q$. Each $U_{V_j} \in G_V$ selects his private key $v_j \in Z_q$ and computes his public key $Y_{V_j}{}' = g^{v_j}\ mod\ p$ and $Y_{V_j}{}'' = g^{V.v_j}\ mod\ p$. Then G_v publishes three group public keys Y_V $Y_V{}'$, $Y_V{}''$ where

$$Y_V = g^V\ mod\ p$$

$$Y_V{}' = \prod_{i=1}^{n} Y_{V_j}{}'\ mod\ p$$

$$Y_V'' = \prod_{i=1}^{n} Y_{V_j}''\ mod\ p$$

Multisignature generation phase:

All signers in G_s perform the following steps to generate the multisignature of a message M for the specified group of verifiers:

1. Each $U_{S_i} \in G_s$ choose a random integer r_i, computes $g^{r_i} mod\ p$ and $(Y_V')^{-r_i} mod\ p$ and sends the pair to U_{S_c} .

2. U_{S_c} checks if $(Y_V')^{-r_\alpha} = (Y_V')^{-r_\beta} (mod\ p)$ for all $\alpha \neq \beta$. If the equality holds then U_{S_c} informs U_{S_α} and U_{S_β} of resending their new pairs; otherwise he broadcasts $\{(Y_V')^{-r_1} (mod\ p), (Y_V')^{-r_2} (mod\ p), \dots, (Y_V')^{-r_n} (mod\ p)\}$ to all signers in G_S.

3. Each $U_{S_i} \in G_s$ computes

$$t_1 = \prod_{i=1}^{n} (Y_V')^{-r_i} (mod\ p)$$

$$t_2 = {t_1}^{-S}\ mod\ p$$

$$C_1 = M.t_1 . (Y_V')^{-t_2}\ mod\ p$$

and $$w_i = r_i - C_1 . s_i\ mod\ q$$

and sends w_i to U_{S_c}.

4. Upon receiving all w_i $(i = 1, 2, \dots, n)$, U_{S_c} verifies the validity of w_i by checking if $g^{r_i} = g^{w_i} . \left(Y_{S_i}'\right)^{C_1} (mod\ p)$ for i= 1, 2, 3,….,n. If all partial signatures are valid then U_{S_c} computes $C_2 = \sum_{i=1}^{n} w_i\ mod\ q$, and sends M and (C_1, C_2) to G_V. .

Multisignature verification phase:

All verifiers in G_V perform the following steps to recover the message from the multisignature and check the message**:**

1. U_{V_c} computes

   $(Y'_S)^{C_1} \, mod \, p$, $(Y_S)^{C_2}(Y''_S)^{C_1} \, mod \, p$ and then sends the results to all verifiers in G_V .

2. Each $U_{V_j} \in G_V$ computes $((Y_S')^{C_1})^{v_j} \, mod \, p$ and $((Y_S)^{C_2}(Y_S'')^{C_1})^{v_j} \, mod \, p$ and sends the results to U_{V_C}

3. U_{V_C} computes

$$t_1{}^{-1} = (Y_V')^{C_2} \cdot \prod_{j=1}^{m} ((Y_S')^{C_1})^{v_j} \, mod \, p$$

$$t_2 = \prod_{j=1}^{m} ((Y_S)^{C_2}(Y_S'')^{C_1})^{v_j} \, mod \, p$$

   And obtains $M = C_1 \cdot t_1{}^{-1} \cdot (Y_V')^{t_2} \, mod \, p$. If the message M is meaningful then M is indeed sent and signed by G_S .

He, showed that the clerk of a specified group of verifiers can alone verify the validity of multisignatures without the help of other verifiers [31]. Therefor this scheme can't be applied for the application of generation of multisignature scheme for specified group of verifiers

5.4 Review of Xie and Yu's Scheme

Xie and Yu pointed out that the multisignature scheme of Laih and Yen is vulnerable to a harmful attack and proposed a new and improved scheme The imp--rovement of Laih and Yen's multisignature scheme can be divided into three phases: the system initialization phase, multisignature generation phase and multisignature verification phase [32].

The system initialization phase

The parameters are almost same as those used in Laih and Yen's scheme. Initially, a trusted center chooses a large prime p, a large prime divisor q such that $q|p-1$, an element g in Z_p of order q, and a one-way hash function $H(\cdot)$. These are then

published as the public parameters. Let $G_s = \{U_{s_1}, U_{s_2}, \dots, U_{s_n}\}$ be the signer group of n signers and $G_v = \{U_{v_1}, U_{v_2}, \dots, U_{v_n}\}$ be the verifier group of m verifiers. In $G_s$and G_v each of them has a special user. Called the'clerk'. Each $U_{s_i} \in G_s$ chooses his secret key $s_i \in Z_{q^*}$ and computes his public key $Y_{s_i} = g^{-s_i} \, mod \, p$. In the same way, each $U_{v_j} \in G_v$ chooses his secret key $v_j \in Z_{q^*}$ and then computes $Y_{v_j} = g^{-v_j} \, mod \, p$ as his public key. G_s's public key $Y_s = \prod_{i=1}^{n} g^{-s_i} \, mod \, p$ and G_v's public key $Y_v = \prod_{j=1}^{m} g^{v} \, mod \, p$ are then published.

Multisignature generation phase :

All signers in G_S perform the following steps to generate the multisignature of message M for the specified group G_V of verifiers.

(1) Each $U_{s_i} \in G_s$ selects a random element $k_i \in Z_{q^*}$ and computes $x_i = {Y_v}^{k_i} \, mod \, p, r_i = g^{k_i} \, mod \, p$ and sends x_i & r_i to U_{s_c} . U_{s_c} computes

$$x = \prod_{i=1}^{n} x_i \, mod \, p$$

$$r = \prod_{i=1}^{n} r_i = g^{\sum_{i=1}^{n} k_i} \, (mod \, p)$$

and broadcasts x and r to all signers in G_s .

(2) Each $U_{s_i} \in G_s$ computes

$$w_i = k_i(r + h(x, M)) - s_i \, mod \, q$$

then sends w_i to U_{s_c}

(3) Upon receiving all $W_i \ (i = 1, 2, \dots, n)$ U_{s_c}verifies each U_{s_i} partial signature by checking

$$r_i^{(r+h(x,M)} = y_{s_i} g^{w_i} \, mod \, p, (i = 1,2, \dots, n)$$

If all of the above equations hold, the multisignature can be obtained as (r, w) where $w = \sum_{i=1}^{n} w_i \, mod \, p$ and U_{s_c} sends it to G_v .

Multisignature Verification phase:

All verifiers in G_v wish to verify the multisignature of message 'M' and do the following steps:

Step 1: Each $U_{v_j} \in G_v$ computes $X_j = r^{-v_j} \, mod \, p$ and sends X_j to U_{v_c}

Step 2: Each U_{v_c} computes $X = \prod_{i=1}^{m} X_j mod \, p$ and broadcasts X to all verifiers in G_v

Step 3: Each U_{v_j} verifies the validity of the multisignature (r, w) for the message M by checking

$$r^{r+h(X,M)} = y_s g^w \, mod \, p$$

5.5 Review of Zhang and Xiao's scheme

In this paper, the authors present a new multisignature scheme for specified group of verifiers. Forging signatures in the proposed scheme is equivalent to forging Harn's signatures [35]. The proposed scheme can withstand He's attack [31].

The procedure of the scheme contains three phases: the key generation phase, the multisignature generation phase and the multisignature verification phase.

Key generation phase.

Let $G_s = (u_{s_1}, u_{s_2}, \ldots, u_{s_n})$ be the group of n signers and $G_v = (u_{v_1}, u_{v_2}, \ldots, u_{v_m})$ be the group of m verifiers. In each group, there is a specified user, called clerk. In each group there is a specific user called clerk. The clerk u_{s_c} of the signers' group is responsible for verifying all partial signature signed by signer in G_s and combining them into multisignature.

The Clerk u_{v_c} of the verifiers' group is responsible for assisting all verifiers in G_v to verify the multisignature. The trusted center plays here a crucial role in performing this multisignature scheme.

The trusted center plays here a crucial role in performing this multisignature scheme. The trusted center T_c selects 2 layers of 'p' & 'q' such that $q \mid p - 1$. It also

selects an element $g \in Z_{p^*}$ with order 'q'. Each of $u_{s_i} \in G_s$ and $u_{v_i} \in G_v$ register themselves with the trusted center in order to become a part of the scheme. T_c distributes private key $S_i \in Z_{q^*}$ to each signer and private key $V_i \in Z_{q^*}$ to each verifier. Each of the signer computes his public key $Y_{s_i} = g^{sp} \, mod \, p$ and similarly each verifier computers his public key $Y_{v_i} = g^{vi} \, mod \, p$. The G_s and G_v respectively publish their group public key Y_s ; where

$$Y_s = \prod_{i=1}^{n} g^{s_i} \, mod \, p$$

$$Y_v = \prod_{j=1}^{m} g^{v_j} \, mod \, p$$

Multisignature Generation Phase:

(a) Each $u_{s_i} \in G_s$ selects a random element $k_i \in Z_{q^*}$ and computes $r_i = g^{k_i} \, mod \, p$ along with $r_i{}' = Y_v{}^{k_i} \, mod \, p$. Then sends r_i & $r_i{}'$ to u_{s_c} .

(b) u_{s_c} computes

$$r = \prod_{i=1}^{n} r_i \, mod \, p$$

$$r' = \prod_{i=1}^{n} r_i{}' \, mod \, p$$

And then sends r' to all signers in G_s .

(c) Each u_{s_i} computes

$$W_i = S_i(H(m) + r') - k_i \, mod \, q$$

Then sends W_i to u_{s_c} .

(d) Upon receiving all $W_i \; (i = 1, 2, \ldots, n)$ u_{s_c}verifies each u_{s_i} partial signature by checking

$$Y_{s_i}{}^{(H(m)+r')} = r_i g^{w_i} \, mod \, p$$

and computes

$$W = \sum_{i=1}^{n} W_i \, mod \, p$$

and sends on and its multisignature (r, w) to G_v .

Multisignature Verification phase:

All verifiers in G_v wish to verify the multisignature of message 'm' and do the following steps:

Step 1: Each $u_{v_j} \in G_v$ computes $x_j = r^{v_j} \, mod \, p$ and sends x_j to u_{v_c}

Step 2: Each u_{v_c} computes $X = \prod_{i=1}^{m} x_j mod \, p$

and broadcasts X to all verifiers in G_v

Step 3: Each u_{v_j} verifies the validity of the multisignature by checking

$$Y_s^{H(m)+X} = r g^w \, mod \, p$$

Later, it was identified that a dishonest clerk of signing group can change the signing message to an arbitrary one while he is cooperating with the signers to produce a multisignature [36]. The weakness is mainly caused by the linear relationship between $H(m)$ and r'in Eq^n1. Further, the vulnerability of Zhang-Xiao's multisignature scheme for specified group of verifiers to forgery attack that an attacker can forge the multisignature for any message was demonstrated [37].

Chapter 6

Conclusion

There are different kinds of multisignature schemes and each of them can be categorized into different categories depending on either the base primitives of the underlying schemes or the role of the signers played during the generation of the multisignature or the verification process of the signatures. We have studied a number of existing multisignature schemes based on DLP. It has been observed that many of the proposed schemes are insure. In general the strength of any signature scheme against forgery depends on the difficulty of finding signer's private key, which in turn depends on the length of the key and underlying base primitive. There is no evidence of any feasible algorithm for efficiently solving DLP yet, but still many security attacks are possible by indirect means [38]. So the size of the key for underlying mathematical problem should be chosen such that the time and cost for forging a multisignature should exceed the value of the information.

Additionally observations revel that some of discussed multisignature schemes involve great computational effort. The requirement that all signers must be present simultaneously to carry out the signature procedure can cause a delay in obtaining a multisignature. In case of structured signature the signing order constraint may cause additional delay. Moreover the fact that OMS demands a particular signing order, users are forced to verify each signer's signature following the inverse order. Sometimes given a group of signers and a multisignature for a given message, the multisignature protocol must be performed once again by all the members of the group each time one new signer joins the group. These reasons make clearly the necessity of a deep study in order to design new more efficient DLP-based multisignature schemes.

Bibliography

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

[2] Intro to Cryptography. Available: http://www.garykessler.net/library/crypto.html.

[3] W. Diffie and M. Hellman, "New directions in cryptography", *in IEEE Trans. Inform. Theory*, vol. IT-30, 1976, pp. 644-654.

[4] R. Rivest et. al., " A method for obtaining digital signatures and public key cryptosystems" *in SIMMONS: Secure Communications and Asymmetric Cryptosystems*,1982.

[5]. US ESIGN Act of 2000, State of WI, National Archives of Australia, The Information Technology Act, 2000.

[6] P.W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring", *Proceedings of the 35th Annual IEEE Symposium on the Foundationsof Computer Science*, 1994, pp. 124-134.

[7] G.J. Simmons, "The Prisoner's Problem and the Subliminal Channel", *in Advances in Cryptology - Crypto '83, Plenum Press*,1984,pp. 51-70.

[8] T. EIGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *in IEEE Trans.Inform. Theory*, vol. IT-31, 1985.

[9] L. Harn, "Digital multisignature with distinguished signing authorities," *in Electronics Letters*, vol. 35, 1999, pp. 294-295.

[10] K. Itakura ,K Nakamura, "A public-key cryptosystem suitable for digital multisignatures", *in NEC Res. Development*, vol. 71,1983 , pp. 1-8.

[11] S. J. Hwang et. al., "A New Digital Multisignature Scheme With Distinguished Signing Authorities," *in J. Inform. Sci. Eng.*, vol. 19(5), 2003, pp. 881-887.

[12] Z. Zhang, G. Xiao, "New multisignature scheme for specified group of verifiers", *in Appl. Math. Comput,*. vol.157, 2004, pp. 425–431.

[13] M. Burmester, et. al. "A structured ElGamal-type multisignature scheme", *in Proceedings of PKC2000, Lecture Notes in Computer Science* , Springer Berlin Heidelberg, 2000, pp.466-483.

[14] L. Harn, T. Kiesler "New scheme for digital multisignatures", *in Elect. Lett.*, vol. 25,1989, pp. 1002-1003.

[15] L. Harn, T. Kiesler "RSA blocking and multisignature schemes with no bit expansion.", *in Elect. Lett.*, vol. 26, 1990, pp. 1490-149.

[16] S. Park et. al., " Two efficient RSA multisignature schemes", *Information and Communications Security*, Springer, Heidelberg, vol. 1334, 1997,pp. 217-222..

[17] L. HARN, "Efficient digital multisignatures" *submitted to EUROCRYFT '94*

[18] C. Paul et. al. , "Handbook of Applied Cryptography". *CRC Press*, 1996.

[19] Pointcheval and Stern,"Security proofs for signature schemes", *In EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT*, 1996.

[20] Wesstein, Eric W. "RSA Encryption." From Mathworld, an online encyclopedia. April, 2001. Available: http://mathworld.wolfram.com/RSAEncryption.html.

[21] Junod, Pascal. "Cryptographic secure pseudo-random bits generation: The Blum-Blum-Shub generator.", 1999.

[22] S. Baral S. Mohanty, B. Majhi, " A novel time-stamped signature scheme based upon dlp*" in 1st International conference on Recent Advances in Information Technology (RAIT),* 2012,pp. 6-10.

[23] Z. C. Li et. al. "Cryptanalysis of Harn digital multi signature scheme with distinguished signing authorities," *in Electr. Lett.,* vol. 36(4), 2000, pp.314-315.

[24] M. N. Hieu and H. D. Tuan, "New Multisignature Schemes with Distinguished Sighing Authorities" *in proc of int. conf. advanced technologies for communication*,2012, pp. 283-288.

[25] Susilo, Willy. "Short fail-stop signature scheme based on factorization and discrete logarithm assumptions." *Theoretical Computer Science*, vol. 410(8) , 2009, pp.736-744.

[26] H. Doi et. al.,"On the security of the RSA-based multisignature scheme for various group structures", *Proceedings of ACISP2000, Lecture Notes in Computer Science*,2000, pp. 352-367.

[27] K. Kawauchi et. al., "Probabilistic Multi-Signature Schemes using a One-Way Trapdoor Permutation ", *IEICE Transactions on Fundamentals*, vol. E87-A, No.5, 2004, pp.1141-1153.

[28] W. Luo, Changying Li "A Structured Multi-signature Scheme Against Forgery Attack", *I.J.Wireless and Microwave Technologies*, vol. 6, 2011, pp. 65-72,

[29] J. ZHANG "Cryptographic Analysis of the Two Structured Multi-signature Schemes", *Journal of Computational Information Systems*, vol. 6(9),2010, pp. 3127-3135.

[30] C.S Laih, S.M. Yen, "Multisignature for specified group of verifiers*", J. Inform. Sci. Engrg.*, vol. 12(1),1996, pp. 143-152

[31] He, W.H., "Weakness in some multisignature schemes for specfied group of verifiers" *in Inform. Proc. Lett.*,vol. 83, 2002, pp. 95-99.

[32] Xie, Qi, and Xiu-yuan Yu. "Improvement of Laih and Yen's multisignature scheme." *Journal of Zhejiang University SCIENCE*, vol. 5(9),2004, pp. 1155-1159.

[33] S.M. Yen, "Cryptanalysis and repair of the multi-verifier signature with verifier specication" *in Computers & Security*,vol. 15(6), 1996, pp. 537-544

[34] S.J. Hwang et. al., " An encryption/multisignature scheme with specified receiving groups", *in Comput.Systems Sci.Engrg.*, vol. 13(2) ,1998, pp.109–112.

[35] L.Harn, " New digital signature scheme based on discrete logarithm" *IEE Electron.Lett.*, vol.30(5), 1994, pp. 396–398,

[36] Lv et. al., "Security of a multisignature scheme for specified group of verifiers.", *Applied mathematics and computation*, vol. 166(1), 2005, pp. 58-63.

[37] Yoon, E.J., Yoo, K.Y, "Cryptanalysis of Zhang-Xiao's multisignature scheme for specified group of verifiers", *Applied mathematics and computation*,vol. 170(1),2005, pp. 226-229.

[38] K. Rabah , "Security of Cryptographic Protocols Based on Discrete Logarithm Problem", *Journal of App. Science*, vol. 5 (9), 2005,pp. 1692-1712.

[39] Digital signature procedure, Available: http://docs.oracle.com/cd/E19316-01/820-2765/images /design.gif.