STRONG PROXY SIGNATURE SCHEME WITH PROXY SIGNER PRIVACY PROTECTION

BY
SHUM KWAN

A THESIS
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF PHILOSOPHY
DIVISION OF INFORMATION ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG
MAY 2002

Acknowledgement

I would like to thank my supervisor Prof. Victor Keh-Wei Wei for his guidance on my research process. It is Prof. Wei who introduced me to the world of cryptography. He provided me the best environment for research. Prof. Wei has provided me the best equipment for my research. This thesis would not have been possible without his valuable ideas, teaching and support.

I have to thank my best friend, Jo who helped me a lot in improving my writing. Also need to thank my dearest classmates, Anthony Chan, who is my classmate since high school, has shared many of ideas with me. Simon Chan, who is my best friend, helped me the mathematics. Brain Lam, who is also my best friend, shared many unforgettable research experience with me.

I would also like to thanks my lab-mates: Ah Wah, Micheal, Mandy, and Ah Tim. They are FYP students in my lab, and shared so much great time in the studying with me.

Last but not least, I have to say thanks to my mother for giving me so much care and support.

Abstract

Proxy signatures allow a principal to delegate its right of signing messages to another principal. Traveling executives can delegate to their secretaries to sign certain documents during their absence. Managers can delegate to their subordinates to perform certain signatures. Delegating the power of digital signature on digital documents to a proxy creates many technical challenges.

Mambo, et al. [MUO96] gave a systematic discussion of proxy signatures. Lee, et al. [LKK01A] constructed a strong non-designated proxy signature scheme in which the proxy signer had strong non-repudiation. Thus far, development of proxy signatures focused on protecting the original signer. The proxy signature schemes in [MUO96, KIM97, LKK01A, KBC00] prevented proxy signers from signing messages unfavourable for the original signers. These schemes also prevented proxy signers to repudiate their signatures.

Anonymity is a desirable property in electronic commerce applications. There may be situations that the proxy signer wishes to remain private. In this thesis, an enhancement to Lee, et al.'s scheme is suggested. We use the certified alias to extents their scheme. The resulting proxy signature scheme protects the privacy of the proxy signers. A detail mathematical description of the our scheme is presented.

With proxy signer privacy protection, many functionalities can be achieved by proxy signatures. This thesis also discuss two potential applications of our proposed proxy signature scheme: *mobile agent* and *group signature*.

Submitted by Shum Kwan
for the degree of Master of Philosophy in Information Engineering
at The Chinese University of Hong Kong, May 2002

摘要

利用現今的數據加密技術, 委託人可以用「代理式簽署」來委託其他人簽署電子文件。「代理式簽署」有很多實際用途, 例如一個上司要出外工幹, 他可以委託他的秘書代他簽署文件。但如何限制受託正確地簽署文件, 一直是一個很具挑戰性的問題。

Mambo 和他的同事在「MUO96」中系統地分析「代理式簽署」。 Lee 和他的同事在「LKK01A」中發表了一個「強化及可轉移的代理式簽署方案」。至今,「代理式簽署」的研究主要集中保護委託人的利益,「MUO96, KIM97, LKK01A, KBC00」這幾篇論文都提出了有效的方法去保護委託人, 防止受託人濫發簽署。「私隱」是一個很受關注的問題, 有很多情況下, 受託人都不願意披露其真正身份。這篇論文的主旨是研究如何用「認證假名」來保護受託人的私隱。文中附有詳細的數式演說我們提出的「保護受託人私隱的代理式簽署方案」。

有了這項新功能,「代理式簽署」可以應用在「移動電子代理」和「組群簽署」上。文中亦會介紹我們的方案如何解決這兩個問題。

Chapter 1

Introduction

For thousands of years, kings, queens and generals relied on efficient communication to govern their countries. At the same time, they were all afraid of falling their messages into the wrong hands. Such incidents could reveal precious secrets to rival nations and betray vital information to opposing forces. The threat of enemy interception motivated the development of codes and ciphers: techniques for disguising a message so that only the intended recipient can read it.

Cryptography has been used for military purposes in history. With the development of e-commerce, cryptography is an enabling technology for protecting privacy and integrity of digital transactions in the cyber space.

Some of the most fundamental cryptographic primitives are symmetric cipher, asymmetric cipher, digital signature, message digest, zero-knowledge proof, and digital certificates. These primitives have been used extensively in constructing security protocols for e-commerce applications.

This thesis focuses on a special kind of digital signature scheme called the proxy signature. The purpose of using proxy signature is to allow a principal to delegate its right of signing messages to another principal.

Thus far, development of proxy signatures focused on protecting the original signer. The protocols of proxy signature schemes in [MUO96, KIM97, LKK01A, KBC00] prevented proxy signer from signing messages unfavourable for the original signer. These schemes also prevented proxy signer to repudiate his signature.

Anonymity is a desirable property in electronic commerce applications. There may be situations that the proxy signer wishes to remain private. However, the proxy signature schemes in [MUO96, KIM97, LKK01A, KBC00] failed to protect the privacy of the proxy signer. This thesis discusses a proxy signature scheme that achieves both anonymity and identifiability of proxy signer.

1.1 Introduction to topic

Proxy signatures allow a principal to delegate its right of signing messages to another principal. Mambo, et al. [MUO96] motivated the need for delegating signature power by the following scenarios:

Scenario 1

Suppose an employee needs to leave his company for a business trip. During his absence, his secretary will take care of his emails. Since the employee's comment is influential in

decision-making, some of the messages should be signed. The question is *how can the secretary digitally sign the message under the employee's name*?

Scenario 2
Suppose the president of a software company digitally signs all its programs under his secret s so as to certify the correctness of the content. The attached digital signature offers a functionality of detecting alterations done to the programs. Because of security concerns, the president does not want to give s to his programmers. Nevertheless, he does not have enough time to sign all the programs his company produces. The questions is *how the programmers can sign their programs with the name of the president, without obtaining the president's secret s* ?

Proxy signatures are signature schemes that capable of solving the problems in scenario 1 and 2. With proxy signature, an original signer can produce multiple signatures in an efficient and parallel fashion, with the help of his proxies.

1.2 What is proxy signature?

Formally, the proxy signature is a signature scheme that allows an original signer to delegate its signing capability to a proxy signer. With the delegation, the proxy signer creates signatures on behalf of the original signer. To verify the proxy signatures, the verifier checks both the proxy-created signatures and the original signer's delegation.

The following example explains how Alice can delegate Bob the right of signing messages.

1. *Delegate:* Alice certifies Bob's public key and warrant. The warrant describes the relation between Alice and Bob, the messages which Bob is allowed to sign, the period of delegation, ... , etc.
2. *Proxy-Sign:* Bob uses the certified key to proxy-sign. The signature is supplemented by Alice's certified warrant to authenticate the delegation relation between Alice and Bob.
3. *Verify:* Victor, the verifier, uses the ordinary digital signature verification algorithm to check the signature created by Bob. Victor also checks the warrant to confirm Alice's agreement on the signed message.

1.3 Terminologies in proxy signature

This section introduces some terminologies which are used extensively in literatures of proxy signature:

- *Proxy Problem* – They are the problems arose by delegating rights. Proxy signature is a solution to a particular kind of proxy problem: delegating the signature rights
- *Original Signer* – An original signer refers to the grantor who grants his power of signing messages to a proxy signer.
- *Proxy Signer* – A proxy signer refers to the grantee who obtains the power of signing messages from an original signer.

- *Delegation* – It refers to the process of granting the power of signing message. At the end of the delegation process, the proxy signer is capable of signing messages on behalf of the original signer.
- *Warrant* – Warrant is a specification issued by the original signer. It restricts the contents and the period of time for the proxy signer to sign. The identities of proxy signers can be listed in the warrant.
- *Consecutive Execution and Direct Form* – They are two different approaches of constructing cryptographic protocols. Consecutive execution achieves its objective by combining cryptographic primitives in a modular fashion. Direct form achieves the same objective by modifying the equations of the cryptographic primitives. Such modifications establish additional functionalities. The approach of direct form usually has better performance than consecutive execution. However, readers would found that the resulting protocol of consecutive execution is easier to understand.

1.4 Levels of delegation

Mambo, et al. [MUO96] gave a systematic discussion of proxy signature. They mentioned three levels of delegation:

1. *Full delegation*: The original signer gives its private key to the proxy signer and the proxy signer uses the private key to sign messages on behalf of the original signer. Verifiers cannot tell whether the signature was created by the proxy signer or by the original signer.
2. *Partial delegation*: The original signer generates a proxy signature key from its private key and gives the proxy signature key to the proxy signer through a secure channel. The proxy signer uses the proxy signature key to sign messages. The proxy signatures in partial delegation are distinguishable from the signatures created by the original signer.
3. *Delegation by warrant*: As described in the previous section, the original signer's warrant restricts the signing capability of the proxy signer. The warrant and the public signature key are certified by the original signer. The proxy signer obtains the warrant form the original signer and uses the corresponding private key to sign. The resulting signature consists of the proxy-signed signature and the certified warrant.

Different levels of delegation serve different security requirements of proxy signatures. The following points should be considered when deciding the level of delegation:

1. Partial delegation and delegation by warrant are more secure than full delegation. This is because their resulting proxy signatures are distinguishable from ordinary signatures.
2. Delegation by warrant can be implemented by consecutive executing ordinary signature schemes.
3. The faster processing speed of partial delegation has an advantage over delegation by warrant.
4. The delegation by warrant is capable of restricting documents to be proxy-signed.
5. Partial delegation does not specify its valid period. As a result, a proxy revocation protocol is needed.

A proxy signature scheme can be constructed by any of these levels. The most appropriate level of delegation should be selected according to the security requirement, message length, and the computation power of both signer and verifier.

1.5 Previous work on Proxy Signature

The idea of proxy signature was discussed in [VAB91, MUO96]. As a sub problem of the "proxy problem", Neuman [BC93] proposed two schemes for delegating signature right: "bearer proxy" and "delegated proxy". Both of them achieved their goals by consecutive execution.

Mambo, et al. [MUO96] suggested the term "proxy signature". They proposed 2 proxy signature schemes: Basic Protocol and Proxy Protected Protocol. Mambo, et al. constructed their schemes using partial delegation and implemented them by direct form equations. They achieved better performance than the schemes proposed in [BC93].

Kim, et al. [KPW97] further modified the schemes in [MUO96]. Their scheme was called partial delegation by warrant. It inherited the efficiency merit of partial delegation and the functional merit of delegation by warrant. With such modification, the original signer gains the power of restricting the signing capability of the proxy signer. As the warrant included the proxy's identity, this can prevent the transfer of proxy power to another party.

The proxy signature scheme of Lee, et al. [LKK01A] did not explicitly include the identity of the proxy signer in the warrant. In particular, the proxy signer could independently generate a proxy signature key from the warrant. Since the identity of proxy signer is not explicitly listed, the warrant of delegation can be transferred to another party. Scheme in both [KIM97] and [LKK01A] exposed the identity of the proxy signer in the signature. This feature ensures the proxy signer cannot repudiate its signature.

The above schemes [MUO96,KPW97,LKK01A] were discrete-log based. There were also proxy signature schemes based on RSA equations [ST97, KBC00, LKK01B]. Sander, et al. [ST97] suggested the use of CEF (Computing with Encrypted Function) to achieve undetachable signature. Kotzanikolaous, et al. [KBC00] realized that suggestion. Lee, et al. [LKK01B] further extended Kotzanikolaous, et al.'s result to support strong non-repudiation by including the proxy signer's public key in the verification equations.

1.6 Our Contributions

This thesis has two contributions. First, it suggested an enhancement to the existing strong non-designated proxy signature scheme, which was proposed by Lee, et al. [LKK01A]. Instead of exposing the identity of proxy signer explicitly, our scheme adds pseudonymity to delegation. The identity of proxy signer is hidden behind an alias. If needed, the alias issuing authority can identify the proxy by the alias.

The second contribution of this thesis is to demonstrate how to apply proxy signature with proxy signer privacy protection to different applications. Examples on mobile agent and group signature are given in the following chapters.

1.7 Thesis Organization

This thesis is organized as follows:

Chapter 1 is an introduction to the thesis. Chapter 2 introduces some useful cryptographic concepts which are used extensively throughout this thesis. An overview of the features that can be achieved by our scheme is presented in Chapter 3. A detail mathematical explanation of our proposed scheme is given in Chapter 4. Practical applications on our proposed scheme is explored in Chapter 5. The concluding remark is given in Chapter 6.

Chapter 2

Background

This chapter introduces some important cryptographic tools and concepts. They are used extensively in the following chapters.

2.1 Digital Signature

Digital signature of a message is actually a number that depends on the private key of the signer and the content of the message being signed. A signature must be verifiable. In a dispute, an unbiased third party should be able to resolve the matter equitably without accessing to the signer's private key.

The following are some basic definitions related to digital signature:

1. A *digital signature* is a data string which associates with a message and with the originating entity.
2. A *digital signature generating algorithm* is a method for producing a digital signature.
3. A *digital signature verification algorithm* is a method for verifying the authenticity of a digital signature.
4. A *digital signature scheme* consists of a signature generation algorithm and an associated verification algorithm.

There are two general classes of digital signature schemes, which can be briefly summarized as the following:

1. *Digital signature schemes with appendix* require the original message as input to the verification algorithm.
2. *Digital signature schemes with message recovery* do not require the original message as input to the verification algorithm. In this case, the original message can be recovered from the signature itself.

2.2 Digital Certificate and CA.

The application of digital signature includes authentication, data integrity, and non-repudiation. One of the most significant applications of digital signatures is the certification of public keys in large networks. Certification is a mean for a certificate authority (CA)--which is a trusted third party--to bind the identity of a user to his public key. Using the digital certificate, other parties can authenticate the public key of a user without the help from the CA.

2.3 Hash Functions

Cryptographic hash functions play a fundamental role in modern cryptography. A hash function takes a message as an input and produces an output. The output is referred to as a *hash-value*, or simply *hash*. More precisely, a hash function h maps strings of arbitrary finite length to strings of fixed length. Hash functions are used for data integrity in conjunction with digital signature schemes. A message is typically hashed first, and then the hash-value is digitally signed.

The following are the basic properties of hash function:

- Compression: h maps an input x of arbitrary finite bit length to an output $h(x)$ of fixed bit length n.
- Ease of computation: given h and an input x, $h(x)$ is easy to compute.

There are also some additional properties of hash function:

- Collision resistance: It is computationally infeasible to find two distinct inputs which hash to the same output.
- One-way: It is computationally infeasible to find an input that hashes to a pre-specified output.

2.4 Bit commitment

Bit commitment protocols are a basic components of many cryptographic protocols. One party, Alice, commits to the other party, Bob, to a bit b, in such a way that Bob has no idea what b is. At the latter stage, Alice can reveal the bit b and Bob can verify that this is indeed the value to which Alice committed. A good way to think about bit commitment is as if Alice writes the bit and puts it in a locked box. Only Alice has the key of the box. She gives the box to Bob. When the time is ripe, she opens the box. Bob knows that the contents were not tampered with since the box was in his possession. Bit commitment protocols have been used for zero knowledge protocols identification schemes, and multi-party protocols. They can also implement Blum's coin flipping over the phone.

By definition, a bit commitment protocol consists of two stages:

1. The *commit stage*: Alice has a bit b to which she wishes to commit to Bob. She and Bob exchange messages. At the end of this stage Bob has some information that represents b.
2. The *reveal stage*: At the end of which Bob knows b.

For all probabilistic polynomial time, all polynomials p, and a large enough security parameter n, the protocol must obey the following:

1. After the commit stage, Bob cannot guess b with probability greater than $\frac{1}{2}+1/p(n)$.
2. Alice can reveal only one possible value. If she tries to reveal a different value, she is caught with probability at least $1-1/p(n)$.

Chapter 3

Brief introduction to Our Result

3.1 A Proxy Signature Scheme with Proxy Signer Privacy Protection

We propose a strong proxy signature scheme which provides proxy signer privacy protection. Our scheme is an extension to the "Strong Proxy Signature Scheme" suggested by Lee, et al. [LKK01A]. The "Strong Proxy Signature Scheme" achieved the following properties:

1. *Strong unforgeability* – Only the designated proxy signer can create a valid proxy signature. The original signer and unauthorized third parties cannot create a valid proxy signature under the name of a particular proxy signer.
2. *Strong non-repudiation* – Once a proxy signer creates a valid proxy signature, it cannot repudiate its signature against anyone. This property prevents proxy signer from taking back its claim.
3. *Verifiability* – The original signer's agreement on the proxy-signed message can be verified from the signature. The signature can be verified by unbiased verifiers using publicly available parameters.
4. *Non-designated* – The warrant of delegation issued by the original signer is transferable among the proxy signers.
5. *Strong identifiability* – Anyone can determine the identity of the proxy signer from a proxy signature.

In property 5: Strong Identifiability, the identity of proxy signer can be identify from its proxy signature. This property has consequences. There may be situations that the proxy signer wishes to remain private. The original signer may also wishes to keep the trust relation between it and its proxy signer private.

We propose a proxy signature scheme that achieves the first 4 properties of Lee, et al's scheme. In addition to that, our scheme achieves the following properties:

6. *Proxy privacy* – the identity of the proxy signer cannot be revealed from the proxy signature and the signed document alone.
7. *Privacy revocation* – If needed, a proper authority can determine the identity of the proxy signer from its signature.

There are certain utilities in property 6 and property 7. Property 6 provides anonymity for proxy signers who wish to remain private. However, property 7 ensures that the proxy's identity can by traced if the need of investigation arises.

In the next chapter, we will give a detail explanation on how to achieve the above 6 properties.

3.2 Applications of Proxy Signature

In this thesis, the following potential applications of our proposed proxy signature scheme are discussed.

1. *Group signature*: The group signature scheme was introduced by David Chaum in [CH91]. The purpose of the group signature is to protect the anonymity of a group member. With group signature, a group member convince the verifier that he belongs to a group, without revealing his actual identity. In case of disputes, a group administrator can determine the identity of the group member from its signature. Our scheme can be considered as a general form of proxy signature. The group administrator plays the role of the alias issuing authority and the original signer, while the group members play the role of proxy signers. Since our proxy signature scheme provides anonymity and identifiability of proxy signer, it satisfies the requirement of group signature. In latter chapter, we will demonstrate how to apply our scheme to construct a group signature scheme with unlimited group size.
2. *Mobile Agent:* Mobile agent signature problem is another potential application of proxy signature with proxy signer privacy protection. Mobile agents are software entities that are able to migrate across different execution environments. They usually carry out goal-oriented tasks automatically on behalf of their owners. Since mobile agents rely on the computation environment of the hosts to execute, malicious hosts can tamper their code and data. This threat raises the problem of revealing the private key of agent owner if the mobile agent has to create digital signatures. Proxy signature can be applied to solve this problem. According to Lee, et al. [LKK01B], an agent owner can delegate his agent which may signs on its itinerary. In the framework of [LKK01B], the proxy signer is referred to the remote host where the mobile agent signs. As mentioned above, the proxy signature scheme of [LKK01B] reveals the proxy signer's identity; the itinerary of the mobile agent can be determined from the proxy signature. When our scheme is applied to mobile agent signature problem, it protects the itinerary privacy of mobile agents.

Applications of proxy signature is not limited to these areas. With proxy signer privacy protection, even more functionalities can be achieved.

Chapter 4

Detail Explanation of Certified Alias and its Application on Proxy Signature

4.1 Introduction

The aim of this thesis is to discuss how to protect proxy signer's privacy. There were papers discussing the protection of signer privacy [CH91]. Their schemes successfully hide the identity of the signer.

It is obvious that if the signer's own key pair is used in creating signatures, verifiers can identify the signer from the signature. Challenges aroused when the signer wishes to remain private, and, at the same time, want his signature to be authentic. In such a situation, an alias issuing authority, which is a trusted third party, is usually involved to generate alias for the signer. The signer uses the alias in place of its identity.

In this chapter, we will give the definition of "Certified Alias", explain how it works with proxy signature.

4.2 Protecting Signer Privacy Using Certified Alias

Alias is an temporary identity replacing the actual identity of a principal. There is an unbiased alias issuing authority , or simply trustee, responsible for aliases management. With a certified alias, the signature created by a particular signer cannot directly linked to the identity of the signer. This is because the identity is difficult to compute from the alias alone. In case of disputes, the alias issuing authority can reveal the identity behind the alias. With such a revelation, third parties can verify the identity protected by the alias . To protect its integrity and authenticity, the alias should be certified by the alias issuing authority. The following are definitions of some important terms:

Definition 4.2.1
Identity: The identity of a principal P is a unique number ID_P representing him.

Definition 4.2.2
Identification token: Identification token of a principal P is a tuple generated by the alias issuing authority. It consists of a random number k_P , which is only known to the authority, and the identity of P. An identification token is donated by

(k_P, ID_P)

Definition 4.2.3

Alias: Alias is a number generated by the trustee. It protects the identity of a principal P. It is generated by a cryptographically secure hash function $h(.)$. The alias of a principal P can be computed by hashing his identification token:

$$h_P = h(k_P, ID_P)$$

The random number is introduced to prevent brute force attack on the alias.

Definition 4.2.4

Certified Alias: A Certified Alias for a principal P is a certificate generated by the alias issuing authority. The certificate contains the alias of P and the public signature key K_p. It is donated by

$$Cert_T(h_P, K_P)$$

To protect the privacy of the signers, we propose using the trustee (T) to generate Certified Alias for each signer (P). The signer use the private key corresponding to the public signature key in the certified alias to sign. Since the signature keys are generate by the trustee, the signature created by these keys will not reveal the identity of the signers.

If disputes occur, the alias issuing authority can reveal the identification token (k_P, ID_P) of the signer in question. Verifiers compares the hash value of the identification token with h_P to confirm the signer's identity.

4.3 Constructing Proxy signature Scheme by Consecutive Execution of Cryptographic Primitives (Scheme CE)

This section describes a proxy signature scheme with proxy privacy protection: Scheme CE. The scheme is constructed by combining cryptographic primitives in a modular fashion.

Assumptions

To ensure the correctness of our scheme, we have to make the following assumptions:

1. Let $h(.)$ be a cryptographically secure hash function.
2. Let

   $$\sigma = sign(m, K^{-1})$$

   be a signature algorithm, and

   $$verify(m, \sigma, K) = T / F$$

   be the corresponding verification algorithm. m is the signed message. σ is the signature. K^{-1} is the private key for signing. K is the public key for verifying the signature.
3. The key pair (public key, private key) of the trustee(T) is

   $$(K_T, K_T^{-1})$$

   the key pair of the original signer (M) is

   $$(K_M, K_M^{-1})$$

   The public key of the trustee and the original signer can be retrieved from public directories.

4. The secure channels between the trustee and the proxy signer , and between the origianl signer to the proxy signer are established by cryptographically secure methods, such as SSL, TLS.

Step 1: Issuing alias

In this step, the trustee generates key pairs and certified aliases for the proxies. We only present how the trustee interact with one of the proxies (P). In real cases, the trustee will repeat this procedure until all the proxies get their key pairs and certificates.

1. The proxy signer sends its identity ID_P to the trustee.
2. Trustee generates a key pair
   (K_{TP}, K_{TP}^{-1})
3. The trustee generates a random number
   $k_P \in_R Z$
   and a hash value
   $h_P = h(k_P, ID_P)$
4. The trustee creates a certificate
   $Cert_T(K_{TP}, h_P)$
5. The trustee sends the following tuple to proxy (P)
   $(Cert_T(K_{TP}, h_P), K_{TP}^{-1})$
   through a secure channel.
6. The trustee records
   (h_P, k_P, ID_P)
   in case privacy revocation is needed.
7. The proxy verifies the certificate by
   $verify(Cert_T(K_{TP}, h_P), K_T)$
8. At the end of this step, the proxy signer P obtains its certified alias. P is convinced that the alias is issued by the trustee.

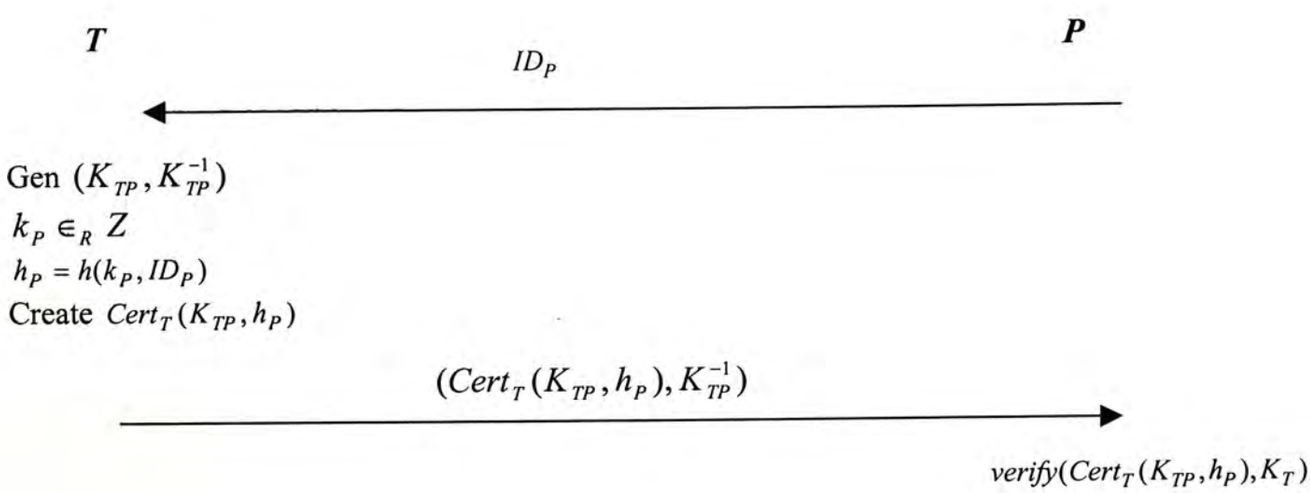

Figure 4.1: Issuing Alias

Step 2: Delegating

In this step, the proxy obtains the delegation from the original signer.

1. The original signer generates a signature on the warrant m_w by computing

$$\sigma_M = sign(m_w, K_M^{-1})$$

2. The original signer then sends the tuple

$$(m_w, \sigma_M, ID_M)$$

to proxy (P).

3. The proxy (P) verifies the signature by

$$verify(m_w, \sigma_M, K_M)$$

4. At the end of this step, the proxy signer can sign on behalf of the original signer.

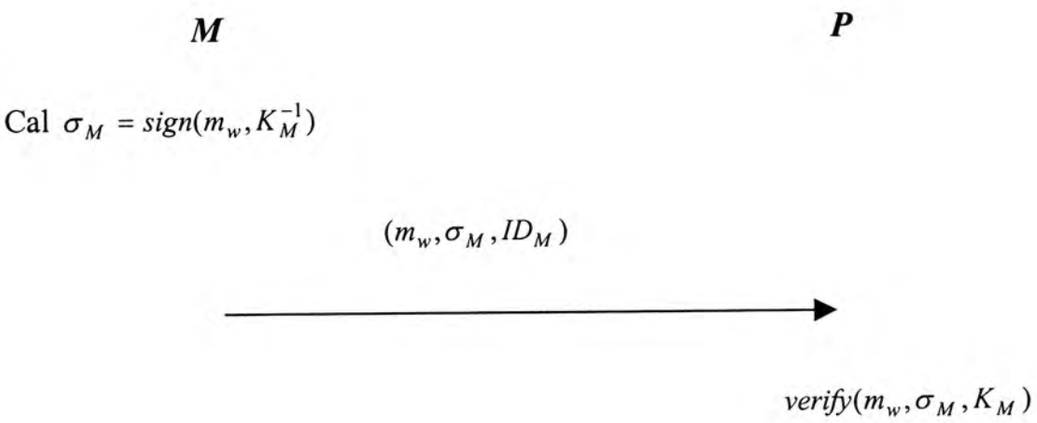

Figure 4.2: Delegating

Step 3: Proxy-Signing and Verifying

1. The proxy (P) signs the message m by

$$\sigma = sign(m, K_{TP}^{-1})$$

2. The proxy produces a proxy signature in the following format

$$(m, \sigma, m_w, \sigma_M, ID_M, Cert_T(K_{TP}, h_P))$$

3. The verifier checks the proxy signature by the follow equations

$verify(m, \sigma, K_{TP})$
$verify(Cert_T(K_{TP}, h_P), K_T)$
$verify(m_w, \sigma_M, K_M)$

4. The verifier checks if the signed message is conform to the warrant m_w.
5. At the end of this step, the verifier accepts the signature.

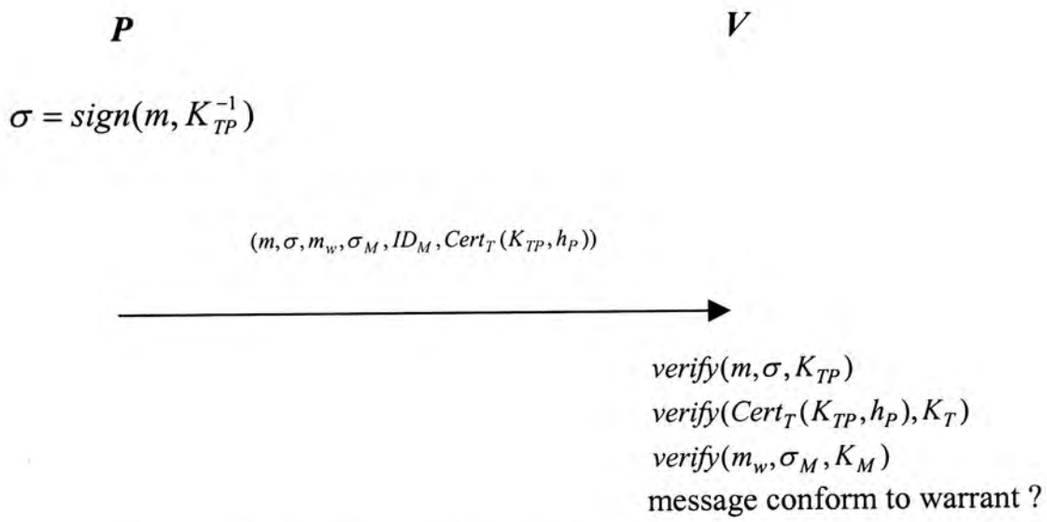

Figure 4.3: Signing and Verifying

Step 4: Revoking Privacy

If the verifier finds the signature is not conform to the warrant, it can ask the trustee to reveal the identity of the proxy signer.

1. The verifier sends the signature
   $(m, \sigma, m_w, \sigma_M, ID_M, Cert_T(K_{TP}, h_P))$
   to the trustee.
2. The trustee verifies that the signature is indeed not conform to the warrant.
3. The trustee sends the identification token
   (k_P, ID_P)
   to the verifier.
4. Verifier compares the hash value
   $h(k_P, ID_P)$
   with h_P to confirm the proxy identity.
5. At the end of this step, the verifier can be convinced that the signature
   $(m, \sigma, m_w, \sigma_M, ID_M, Cert_T(K_{TP}, h_P))$
   is created by the proxy signer P.

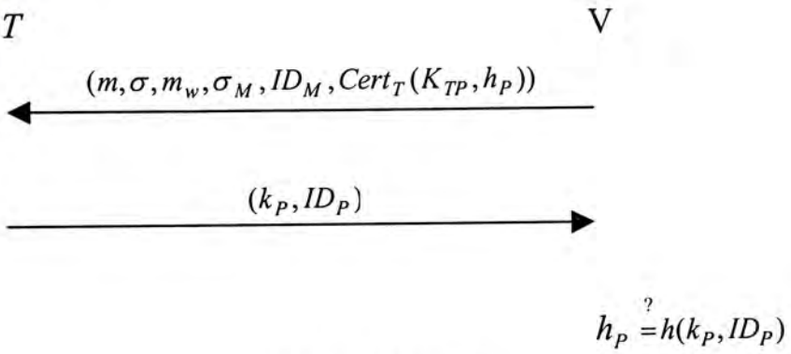

Figure 4.4: Revoking Privacy

Security analysis of scheme CE

1. **Strong unforgeability**: Since the signature is created by the private proxy signature key K_{TP}^{-1}, and only P received them. Original signer and other third parties do not have proxy signature key, so they cannot create valid proxy signature with the name of P.
2. **Strong non-repudiation**: The signature contains the alias of P which is certified by the alias issuing authority. Once the trustee revokes P's privacy, P cannot repudiate his signature.
3. **Verifiability**: Verifier has to check the authenticity of the warrant. If the message is conform to the signed warrant, M's agreement on the signed message is verified.
4. **Non-designated**: In the step of delegation, the original signer only sends (m_w,σ_M,ID_M) to the proxy signer. This signed warrant does not specify the identity of the proxy signer. As a result, the delegated proxy signer can pass the warrant to another third party.
5. **Proxy privacy**: Since the public proxy signature does not contain any information directly related to identity of the proxy signer, verifiers cannot determine the proxy signer's identity from the signature alone.
6. **Privacy Revocation**: The alias issuing authority keeps the record of (h_P,k_P,ID_P) for every alias. If disputes arises on the alias h_P, the trustee can reveal the identification token (k_P,ID_P). After the identification token is revealed, anyone can verify the hash value of (k_P,ID_P) with h_P.

4.4 Constructing Proxy signature Scheme by Direct Form Equations (Scheme DF)

In this section, we present a more efficient scheme constructed by direct form: Scheme DF. The main difference between Scheme DF and Scheme CE is that generation of proxy signature key. In Scheme DF, the proxy signer combines the private part of the certified alias and the private part of certified warrant to form the private signature key. This feature provides extra security merit over Scheme CE.

Assumptions:

To ensure the correctness of our scheme, we have to make the following assumptions:

1. Let p be a prime number. g is an element in the multiplicative group Z_p^* whose order, q, is a large prime dividing p-1. A discrete logarithm problem with inputs p, g and $a \in_R Z_q^*$ consists of finding the $b \in Z_p^*$ such that $g^a \equiv b \bmod p$. The discrete logarithm problem is hard to solve.
2. Let $h(.)$ be a cryptographically secure hash function.
3. Let $\sigma = sign(m,s)$ be a DL-based signature algorithm where m is the message to sign and s is the private key of the signer. σ is the created signature.
4. Let $verify(m,\sigma,v)$ be the DL-based signature verification algorithm, which return true or false. m is the signed message, σ is the signature, and v is the public key of the signer.
5. The key pair (private key, public key) of Trustee (T) is
   (x_T, y_T)
   such that $x_T \in Z_q^*, y_T \equiv g^{x_T} \bmod p$.
   The key pair of Original signer (M) is
   (x_M, y_M)
   The public keys of the trustee and the original signer can be retrieved from public directories.
6. The secure channels between the trustee and the proxy signer , and between the origianl signer to the proxy signer are established by cryptographically secure methods, such as SSL, TLS.

Step 1: Issuing Alias

In this step , the trustee sends certificates for each proxies. We only present how the trustee interact with one of the proxies (P). The trustee will repeat this procedure until all the proxies get their certificates.

1. The proxy signer sends its identity ID_P to the trustee.
2. The trustee generate a random number $k_P \in_R Z$
3. The trustee generate an alias by
   $h_P = h(k_P, ID_P)$
   where ID_P is the identity of the proxy signer. h_P is the alias of the proxy signer.
4. The trustee then signs h_P by computing
   $k_T \in_R Z_q^*$
   $r_T = g^{k_T} \bmod p$
   $s_T = x_T h(h_P, r_T) + k_T \bmod q$
5. The trustee sends the following triple to the proxy signer through a secure channel
   (h_P, r_T, s_T)
6. The trustee records
   (h_P, k_P, ID_P)
   in case of privacy revocation.
7. The proxy signer checks the congruence to verify the signature:
   $g^{s_T} \stackrel{?}{=} y_T^{h(h_P, r_T)} \cdot r_T \bmod p$
8. At the end of this step, the proxy signer obtains a certified alias
   (h_P, r_T, s_T)

from the trustee.

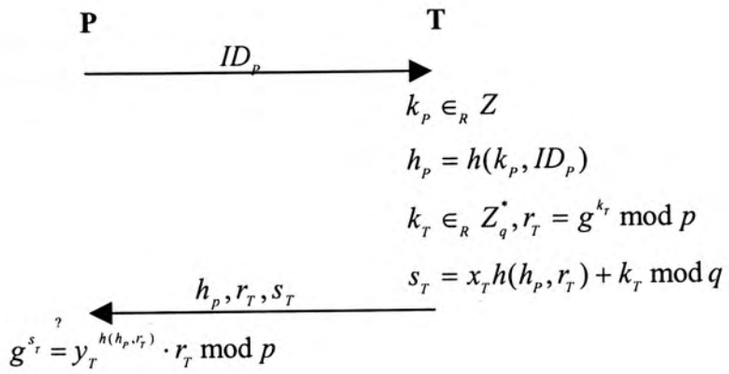

Figure 4.5: Issuing Alias

Step 2: Delegating

When original signer is ready to delegate its signing capability,

1. The original signer signs the warrant m_w by computing

$$k_M \in_R Z_q^*$$
$$r_M = g^{k_M} \bmod p$$
$$s_M = x_M h(m_w, r_M) + k_M \bmod p$$

2. The original signer then sends the tuple

$$(m_w, r_M, s_M)$$

to the proxy signer.

3. The proxy signer verifies the signature by checking the congruence:

$$g^{s_M} \stackrel{?}{=} y_M^{h(m_w, r_M)} \cdot r_M \bmod p$$

4. At the end of this step, the proxy signer obtains a certified warrant

$$(m_w, r_M, s_M)$$

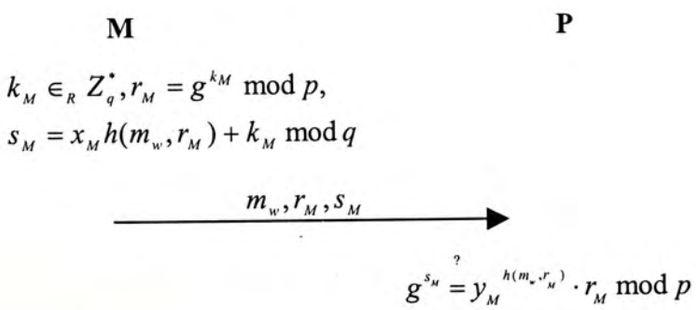

Figure 4.6: Delegating

Step 3: Proxy-signing and Verifying

1. The proxy signer generates a private proxy signature key by computing

$$s = s_M + s_T \bmod q$$

2. The proxy signs the message m by DL-based signature algorithm, using s as private key.

$$\sigma = sign(m, s)$$

3. The proxy signer produces a signature

$$(m, \sigma, r_M, m_w, ID_M, h_P, r_T)$$

4. The verifier computes the public proxy signature key by

$$y = y_M^{h(m_w, r_M)} \cdot r_M \cdot y_T^{h(h_P, r_T)} \cdot r_T$$

5. The verifier verifies the signature by

$$verify(m, \sigma, y)$$

7. The verifier check if the message is conform to the warrant.
8. At the end of this step, the verifier accepts the proxy signature.

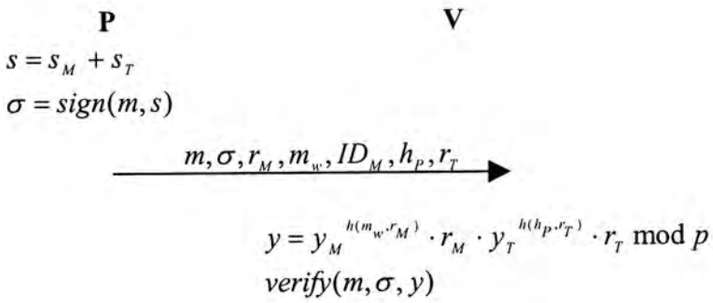

Figure 4.7: Signing and verifying

Step 4: Revoking Privacy

If the verifier finds the proxy signature is not conform to the warrant, it can ask the trustee to reveal the identity of the proxy.

1. The verifier send the proxy signature

$$(m, \sigma, r_M, m_w, ID_M, h_P, r_T)$$

to the trustee

2. The trustee verifies that the signature is indeed not conform to the warrant
3. The trustee sends the identification token

$$(k_P, ID_P)$$

to verifier.

4. Verifier compares the hashed value

$$h(k_P, ID_P)$$

with the alias to confirm the proxy identity.

5. At the end of this step, the verifier can be convinced that the signature

$$(m, \sigma, r_M, m_w, ID_M, h_P, r_T)$$

is created by the proxy signer P.

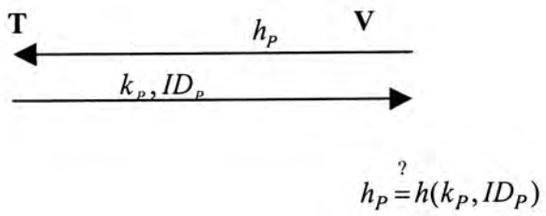

Figure 4.8: Revoking Privacy

Security analysis of scheme DF

1. **Strong unforgeability**: Since the proxy signature key is generated using s_M and s_T, and only P receive them. Original signer and other third parties do not have proxy signature key, so they cannot create valid proxy signature with the name of P.
2. **Strong non-repudiation**: The signature contains the alias of P which is certified by the alias issuing authority. P cannot repudiate his signature once the trustee revokes P's privacy.
3. **Verifiability**: The public proxy signature key includes the term $y_M^{h(m_w, r_M)} \cdot r_M$ which represents M's agreement on m_w. If the message is conform to the signed warrant, M's agreement on the signed message is verified.
4. **Non-designated**: In the step of delegation, the original signer only sends (m_w, r_M, s_M) to the proxy signer. Since this signed warrant does not specify the identity of the proxy signer, the delegated proxy signer can pass the warrant to another third party.
5. **Proxy privacy**: Since The public proxy signature does not contain any information directly related to identity of the proxy signer, the verifiers cannot determine the proxy signer's identity from the signature alone.
6. **Privacy Revocation**: The alias issuing authority keeps the record of (h_P, k_P, ID_P) for every alias. If disputes occur on the alias h_P, the trustee can reveal the identification token (k_P, ID_P). After the identification token is revealed, verifiers can compare the hash value of (k_P, ID_P) with h_P to confirm the proxy singer's identity.

4.5 Comparison between scheme CE and scheme DF

Scheme DF is more efficient than Scheme CE because only one signature verification is needed . Scheme CE requires 3 signature verifications. In scheme DF, computing y involves 2 exponential operations. Although more exponential operations are involved, [MUO96] showed

that direct form equations have still performance advantages over consecutive execution. Simulation details and complexity analyses are contained in [MUO96].

Scheme DF has another advantage over scheme CE. For the same proxy signer, different proxy signature keys are used for different original signers. Scheme CE use the same proxy signature key for all original signers. The problem of using one proxy signature key for all delegation is that a third party C may manipulate the signatures. Lets consider the following example:

> $A1$ and $A2$ delegates P to sign messages for them. P signs a message $m1$ for $A1$, $m2$ for $A2$, producing the following signatures
> $(m_1, \sigma_1, m_{w1}, \sigma_{A1}, ID_{A1}, Cert_T(K_{TP}, h_P))$
> $(m_2, \sigma_2, m_{w2}, \sigma_{A2}, ID_{A2}, Cert_T(K_{TP}, h_P))$
>
> Both of the signature are valid signature. Yet if malicious third party C can manipulate the signatures by producing a new signature $(m_2, \sigma_2, m_{w1}, \sigma_{A1}, ID_{A1}, Cert_T(K_{TP}, h_P))$. As long as the message m_2 conform to warrant of m_{w1}, verifiers will think P has sign m_2 on behalf of $A1$. P cannot convince verifiers he has not created such a signature.

Scheme DF will not suffer from this problem, because the proxy signature key is generated on each delegation.

Scheme CE has an advantage over scheme DF. It does not limit the signature and verification algorithm to be DL-based. Both RSA-based and DL-based public key cryptosystem can be applied in Scheme CE.

4.6 Chapter Summary

In this chapter, we have described the concept of certified alias. We have presented how to apply certified alias to construct a proxy signature scheme that protects proxy signer privacy. We have also suggested two practical implementation of our proxy signature scheme: Scheme CE and Scheme DF.

Chapter 5

Applications of Proxy Signature with Proxy Signer Privacy Protection

In previous chapter, we have suggested 2 proxy signature schemes with proxy signer privacy protection. In this chapter, we demonstrate some applications of our proxy signature scheme.

5.1 Secure Mobile agent Signature with Itinerary Privacy

5.1.1 Introduction to Mobile Agent

Mobile agents are goal-directed programs. They are capable of suspending themselves on one platform and moving to another platform during execution.

There has been a number of models describing mobile agent systems. For discussing security issues, it is sufficient to use a simple model which contains only two components: the *agent* and the *host*. The agent comprises the code and state information for computation. The computation is goal-directed, and is performed automously by the agent on behalf of the agent owner with the cooperation of other agents. Mobility allows an agent to move among hosts. The hosts provides the computational environment in which the agent operates. For simplicity, the host itself can be thought of as a collection of special purpose agents. These agents work together to provide the services and computational environment for mobile agents.

5.1.2 Review on Lee, et al. strong non-designated proxy signature scheme for mobile agents

Since mobile agents rely on the resources of hosts to execute, all code and data are exposed to the hosts. Malicious hosts can spy on the data and state information of the mobile agents. If traditional digital signature schemes were used, the mobile agents must carry their owner's private keys. As a result, the private keys are subjected to read attack by malicious hosts. Lee, et al. [LKK01B] discussed how to applied proxy signature to a mobile agent. This section reviews the scheme of Lee, et al. [LKK01B].

In the framework of mobile agent suggested by Lee, et al., there are 4 entities:

Agent owner (A) : performs the role of original signer in proxy signature. It delegates the signing capability to the mobile agent.

Host (B) : carries out proxy signature on behalf of the agent owner, and thus performs the role of proxy signer.

Agent : acts as a message carrying vehicle. It carries the proxy signature keys, warrant of agent owner and created signatures on its itinerary.
Verifier (V): verifies the proxy signature created by the host.

The scheme of Lee, et al. contains 3 steps

Step 1: Preparing the agent (by agent owner)

The agent owner generates a signature on the warrant m_w by computing

$$k_A \in_R Z_q^*,$$

$$r_A = g^{k_A} \bmod p$$

$$s_A = x_A \cdot h(m_w, r_A) + k_A \bmod q$$

and load the tuple (m_w, r_A, s_A) to the mobile agent.

Step 2: Executing the agent (by host)

When the agent arrived at the host, and ready to create proxy signature, it generates a proxy signature key (x_P, y_P) by computing

$$x_P = s_A + x_B \bmod q,$$

$$y_P = g^{x_P} \bmod p$$

where (x_B, y_B) is the static key pair of the host. And the host is strongly motivated to keep the static private key secret.

The host uses the generated proxy signature key to sign message :

$$\sigma_P = sign(m, x_P)$$

and producing the signature as

$$(m, \sigma_P, r_A, m_w, ID_A, ID_B)$$

Step 3: Verifying the signature (by verifier)

The verifier check the signature by

$$verify(m, \sigma_P, y_P),$$

$$where\ y_p = y_A^{h(m_w, r_A)} r_A y_B \bmod p$$

It also check if the signed message is conform to the warrant of the agent owner.

Remarks on Lee, et al. scheme

Since the resulting signature $(ID_A, m_w, ID_B, m, r_A, \sigma_P)$ contains the identity of proxy signer. The verifiers can determine where the agent created the signature since the proxy signer is the host generated the signature. In the next section , we try to solve this problem by using proxy signature with proxy signer privacy protection.

5.1.3 Constructing Signature scheme for Mobile Agent using Proxy signature with Proxy Signer Privacy Protection

The scheme of Lee, et al. [LKK01B] successfully prevents malicious host from reading the agent owner's private key. However, the resulting signature reveals the identity of the proxy signer. Verifiers can determine the host on which the mobile agent created the signature. This property offences itinerary privacy of a mobile agent. In this section, we give a construction of

signature scheme for mobile agent based on proxy signature with proxy signer privacy protection. The resulting scheme has all the properties of the scheme of Lee, et al. . Our scheme also achieve itinerary privacy protection for mobile agent.

Scheme DF in chapter 4 is applied directly. We follow the framework defined by Lee, et al. [LKK01B]. In addition to that, we add an entity trustee to the framework.

The resulting scheme consists of 4 steps:

Step 0: Setup

In this component, the trustee generates certificate for each of the hosts. Here we only show the interaction between the trustee and one of the hosts (B). The trustee repeats this step until all the hosts get their certificates.

The trustee generates an alias for host B by computing

$$k_B \in_R Z,$$
$$h_B = h(k_B, ID_B)$$

The trustee the signs the alias by its key pair (x_T, y_T)

$$k_T \in_R Z_q^*,$$
$$r_T = g^{k_T} \bmod p,$$
$$s_T = x_T \cdot h(h_B, r_T) + k_T \bmod q$$

The trustee then sends the following tuple to the securely

$$(h_B, r_T, s_T)$$

Step 1: preparing the agent (by the agent owner)

This component remains unchanged from the scheme of Lee, et al.

Step 2: executing the agent (by the host)

When the agent arrived at the host, and ready to create proxy signature, it generates a proxy signature key (x_P, y_P) by computing

$$x_P = s_A + s_T \bmod q,$$
$$y_P = g^{x_P} \bmod p$$

The host uses the generated proxy signature key to sign message

$$\sigma_P = sign(m, x_P)$$

and produce the signature as

$$(m, \sigma_P, r_A, m_w, ID_A, h_B, r_T)$$

Step 3: Verifying the signature (by verifier)

The verifier checks the signature by

$$verify(m, \sigma_P, y_P),$$
$$where\ y_p = y_A^{h(m_w, r_A)} r_A \cdot y_T^{h(h_B, r_T)} \cdot r_T \bmod p$$

It also check if the signed message is conform to the warrant of the agent owner.

At the end of this step, the verifier accepts the signature.

5.1.4 Remarks

Our scheme skip the use of host static key. Instead, we use the private part the certificate to replace the host static key. Since this part is only known to the host, it can be considered as a trustee generated private key. In the generated signature, the identity ID_B is replaced by the alias h_B and the public part of the certificate r_T. This replacement hides the identity of the signature

creating host. Since the alias is signed by the trustee, the verifier can request the trustee to reveal the protected identity in case of disputes.

5.2 Group Signature with Unlimited Group Size

5.2.1 Introduction to group signature

Group signature was discussed by David Chaum in [CH91]. He introduced it by the following problem:

> A company has several computers, each connected to the local network. Each department of that company has its own printer (also connected to the network) and only the staff of that department are allowed to use their department's printer. Before printing, the printer must be convinced that the staff is working in that department. At the same time, the company wants privacy; the staff's name may not be revealed. If, however, someone uses the printer too often, the director must be able to discover who misused that printer, to send him a bill.

Group signature is a solution to the problem raised by David Chaum. It is a generalization of the membership scheme, in which a group member can convince a verifier that he belongs to a group, without revealing his identity. Formally, the group signature has following properties:

1. Only members of the group can sign messages.
2. The receiver of the signature can verify that it is a valid signature from the group.
3. The receiver of the signature cannot determine which member of the group is the signer.
4. In case of a dispute, the signature can be "opened" to reveal the identity of the signer.

5.2.2 Constructing group signature scheme using certified alias

Proxy signature with proxy signer privacy protection can be applied to construct group signature scheme with unlimited group size. The basic requirements of group signature are: unforgeability, verifiability, anonymity and identifiability [CH91]. In chapter 4, we demonstrated how to achieve both anonymity and identifibility by certified alias. Same technique can be applied on group signature to achieve the desired result.

The resulting group signature will have the following properties:

1. Unforgeability
2. Verifiability
3. Group member privacy
4. identifiability
5. Unlimited group size

A detail description on the group signature scheme with unlimited group size is presented below.

Assumptions:
We follows the results from scheme DF of chapter 4, so the assumptions is the same of scheme DF.

Entities:

The group signature scheme involves 3 entities: Group Administrator (A) , group member (M) and verifier (V). There is at least one group administrator in each group of users. The group administrator is responsible for registering new user to the group. It generates aliases for group members. Each group member use its alias to identify itself in the group. Only the group administrator know the actual identity of each group member. As a result, when investigation is needed, only the group administrator can identify the signer from the group signature.

The group signature scheme with unlimited group size contains 3 steps:

Step 1: Joining Group

When a user (M) wants to join a group (G) , he is required to convince the group administrator (A) that he is a qualified to join G . The authentication of group member can achieve by ordinary authenticate schemes.

After authentication, the group administrator and the group member carry out the following computations:

1. The group member generates a public-key/private-key key pair (x_M, y_M) such that

   $$x_M \in_R Z_q^*$$

   $$y_M = g^{x_M} \bmod p$$

2. The group member sends the public key to the group administrator.
3. The group administrator generates a random number k_M and hashes it with the identity of the group member. The hash value, h_M , is the alias for the group member.

   $$k_M \in_R Z,$$

   $$h_M = h(k_M, ID_M)$$

4. The group administrator then signs the public key and the alias together. The certificate also contain a message, m_w, which states that the alias was issued to a member of group G.

   $$k_A \in_R Z_q^*$$

   $$r_A = g^{k_A} \bmod p$$

   $$s_A = x_A \cdot h(y_M, h_M, m_w, r_A) + k_A \bmod q$$

5. The generated certificate, $(y_M, h_M, m_W, r_A, s_A)$,is then send back to the group member.
6. The group administrator records the tuple $(y_M, h_M, m_W, ID_M, k_M)$.

Step 2: Signing and verifying

1. When the user wants to signs a message, he uses his private key to sign. He creates the signature

   $$\sigma = sign(m, x_M)$$

   He also attach the certificate to the signature he created and produces the group signature

   $$(m, \sigma, y_M, h_M, m_W, r_A, s_A)$$

2. When the verifier wants to verify the signature was created by someone belongs to group G, he verifies the public key in the certificate by

   $$g^{s_A} \stackrel{?}{=} y_A^{h(y_M, h_M, m_W, r_A)} \cdot r_A \bmod p$$

   to confirm it was approved by the group administrator.
3. The verifier then verifies the signature

$verify(m, \sigma, y_M)$

Step 3: Investigating

1. In case of a dispute, the group administrator can identify the corresponding signer of the group signature by searching his database for the tuple
   $(y_M, h_M, m_W, ID_M, k_M)$
2. The group administrator send
   (ID_M, k_M)
   to the verifier.
3. The verifier computes
   $h(k_M, ID_M)$
   and compares the result with the alias in the group signature. If they are the same, the verifier can be convinced that the signer of the group signature is M.

5.2.3 Security Analysis

1. **Unforgeability**: Only the group member can sign with the name M. This is because the group administrator and third parties do not have the private key x_M.
2. **Verifiability**: The verifier can be convinced that the group signature is created by a valid group member. This is because the public key is certified by the group administrator, and the signature is verifiable with the public key by the equation
   $verify(m, \sigma, y_M)$
   These two conditions hold only if the signature is actually created by a valid group member.
3. **Group Member Privacy**: Since the group signature
   $(m, \sigma, y_M, h_M, m_W, r_A, s_A)$
   does not contain any information directly related to the identity of M. The group member's privacy is protected.
4. **Identifiability**: With the help of the group administrator, the tuple
   $(y_M, h_M, m_W, ID_M, k_M)$
   can be identified from the information provided by the group signature. The verifier can know the identity of the signer. The verifier can computes
   $h(k_M, ID_M)$
   and compares the result with the alias in the group signature.
5. **Unlimited group size**: The group administrator do not have to determine how many members does the group have before registering members. As a result, the group size is unlimited.

5.2.4 Remarks

Proxy signatures with proxy signer privacy and group signatures has the following similarities:

1. In both schemes, the proxy (the group member) has to get the authorization from the original signer (the group administrator) before it can sign.

2. The signature created by any of the proxies (any member in the same group) should means the same to the verifier. If dispute does not occur, there is no need to identify which proxy (group member) created the signature.
3. Both schemes should have identifiability and anonymity.

As a result, the group signature scheme we presented in this section actually transform the proxy signer to the group member in group signature. The trustee and original signer to form the group administrator.

5.3 Chapter Summary

The proxy signature with proxy signer privacy protection has many applications. In this chapter, we suggested two practical applications for it. Secure mobile agent signature directly applies the proxy signature scheme. In the group signature scheme with unlimited group size, the group administrator plays the role of both the original signer and the trustee in proxy signature. The techniques developed in our proxy signature scheme are suitable for applications which requires the signer's anonymity and identifiability.

Chapter 6

Conclusions

This thesis focuses on the privacy of proxy signer in proxy signature schemes. The solution to this problem is the certified alias. The thesis starts by introducing the concept of the proxy signature. The definitions and terminologies of proxy signature are given. Previous works on proxy signature are reviewed.

The focus of this thesis is to achieve both anonymity and identifiability of proxy signers. These apparently contradicting objectives are achieved by introducing an alias issuing authority, or, simply, trustee. The trustee generates aliases for proxy signers. The proxy signers use the aliases to protect their privacy. The alias authority also keep track of all the aliases. If disputes occur, the alias authority have the power to reveal the identity of the proxy signer protected by the alias.

We survey the application of proxy signatures. When the property of proxy signer privacy is introduced to proxy signature, the resulting signature scheme is suitable for many new applications. We explorer how it can be applied to mobile agents signature and group signature with unlimited group size.

Above all, proxy signature is still a new kind of cryptographic primitives that can be useful in building many e-commerce applications. Much research and enhancements could be done on the security and application of proxy signatures.

Appendix: Paper derived from this thesis

K. Shum and Victor. K. Wei, "Strong Proxy Signature Scheme with Proxy Signer Privacy Protection." Accepted for publication in *IEEE Eleventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, June, 2002

Bibliography

[VAB91] V. Varadharajan, P. Allen, and S.Black , "An analysis of the proxy problem in distributed systems." *Proc 1991 IEEE Computer Society Symposium on Research in Security and Privacy* (1991) 255-275.

[BC93] B.C Neuman, "Proxy-based authorization and accounting for distributed systems." *Proc 13th International Conference on Distributed Computing Systems* (1993) 283-291

[MUO96] M. Mambo, K. Usuda, E.Okamoto, "Proxy Signature: Delegation of the Power to Sign Messages", *IEICE Trans. Fundamentals*, **E79-A:9** (1996) 1338-1353

[KPW97] S.Kim , S.Park, and D.Won , "Proxy Signature, revisited", *Proc of International Conference on Information and Communication Security* (1997) 223-232.

[KIM98] Seungjoo Kim, *Improved Privacy and Authenticity in Digital Signatures/Key Management.* Ph.D thesis , Division of information Engineering , The Sungkyunkwan University (1998).

[PH97] H. Petersen and P. Horster, "Self-certified Keys Concepts and Applications", *In Proc. Communications and Multimedia Security* (1997) 102-116.

[LKK01A] B. Lee, H.Kim, K.Kim, "Strong Proxy signature and its Applications." *Proc of SCIS (*2001) 603-608.

[KBLB01] H.Kim, J. Baek, B.Lee, K.Kim, "Computing with Secrets for Mobile agent using one-time proxy signature." *Proc of SCIS* (2001) 845-850.

[LKK01B] B. Lee, H.Kim, K.Kim, "Secure Mobile Agent using Strong Non-designated Proxy Signature." *Australasian Conference on Information Security and Privacy* (2001) 474-486

[ST97] T. Sander, C. Tschudin, "Towards Mobile Cryptography", *Technical Report 97-409, International Computer Science Institute, Berkeley* (1997).
http://www.icsi.berjely.edu/~sander/publiccations/tr-97-049.ps

[ST98] T.Sander, C. Tschudin , "Protecting Mobile Agents Against Malicious Hosts." *In G.Vigna, editor, Mobile Agents and Security. Springer-Verlag Verlin Heidelberg* (1998).
http://www.icsi.berkeley.edu/~sander/publications/MA-protect.ps

[KBC00] P.Kotzanikolaous, M. Burmester and V. Chrisskopoulos, "Secure Transactions with Mobile Agents in Hostile Environments." Proc. *ACISP* 2000 (289-297).

[KZ96] K.Zhang , "Nonrepudiable Proxy Signature Scheme", *manuscript* (1997).
http://citeseer.nj.nec.com/360090.html

[KZ97] K.Zhang , "Thresold proxy signature schemes", *Information Secuirty Workshop , Japan* (1997) 191-197.
http://citeseer.nj.nec.com/zhang97threshold.html

[LHW98] Narn-Yih Lee, Tzonelih Hwang, Chih-Hung Wang , "On Zhang's Nonrepudiable Proxy Signature Schemes." *ACSIP* (1998) 415-422.

[CC93] Claude Crépeau, "Cryptographic Primitives and Quantum Theory." http://citeseer.nj.nec.com/127162.html

[MN01] Moni Naor, "Bit Commitment Using Pseudo-Randomness." *Journal of Cryptography* **V2:2** (1991) 151-158.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. "Pseudonym Systems." *Sixth Annual Workshop on Selected Areas in Cryptography (SAC '99)*, **1758** (2000).

[MOV96] Alfred Menezes, Paul C. van Oorschot, Scott A. Vanstone, "Handbook of Applied Cryptography." *CRC Press* (1996).

[BS96] Bruce Schneier , "Applied Cryptography: Protocols, Algorithms and Source Code in C." *Wiley* (1996)

[SS99] Simon Singh, "The Code Book." *A division of Random House Inc,* (1999).

[CH91] D. Chaum and E. van Heijst,"Group signatures." *Advances in Cryptology - Eurocrypt*, (1991) 257-265.

[WAJ99] W. A. Jansen, "Mobile Agents and Security.", *NIST Technical Report, National Institute of Standards and Technology* (1999).

[JW96] Jim White, "Mobile Agents. General Magic White Paper." *General Magic* http://www.genmagic.com/agents

[CHK94] David Chess, Colin Harrison, Araon Kershenbaum, "Mobile Agents: Are They a Good Idea ?" *IBM Technical Report, T.J. Watson Research Center,* (1994).

[JK00] Wayne Jansen, Tom Karygiannis, "NIST Special Publication 800-19 – Mobile Agent Security." *NIST Technical Special Publication* (2000).

[WS98] Uwe G. Wilhelm, Sebastian Staamann, and Levente Buttyn. "Protecting the itinerary of mobile agents." *In 4th ECOOP Workshop on Mobility: Secure Internet Mobile Computations* (1998).

[AF89] M.Abadi and J. Feigenbaum , "On Hiding Information from an Oracle." *Journal of Computer Science*, **39** (1989) 21-50

[AF90] M.Abadi and J. Feigenbaum , "Secure circuit evaluation." *Journal of Cryptology*, **2:1** (1990) 1-12.

[CSV93] Don Coppersmith, Jacques Stern, and Serge Vaudenay. "Attacks on the birational permutation signature schemes." *In Douglas R.Stinson, editor, Proceedings of CRYPTO 93*, **773** (1993) 435-443.

[CMM01] C.M. Mo, *Securing Mobile Agent in Hostile Environment.* M.Phil thesis, division of information Engineering , The Chinese University of Hong Kong (2001)

[SKN00] S.K. Ng , *Protecting Mobile Agents Against Malicious Hosts,* M.Phil thesis, division of Information Engineering , The Chinese University of Hong Kong (2000)