Specifiable Proxy Signature Schemes

Hwang, Shin-Jia and
Department of Information Management,
Chaoyang University of Technology
Wufeng, Taichung Country, 413, Taiwan,
R.O.C.
Email: sjhwang@mail.cyut.edu.tw

Shi, Chi-Hwai*
Department of Information Management,
Chaoyang University of Technology
Wufeng, Taichung Country, 413, Taiwan,
R.O.C.
*Email: s8714607@mail.cyut.edu.tw

Abstract

Since Mambo et al. proposed the concept of proxy signatures in 1996, many proxy signature schemes were proposed. Two of the proposed schemes integrated the specification of the proxy period into the proxy signature scheme. Besides the proxy period, there is some other important specification of the proxy. Therefore, we propose a basic scheme providing the verifiable mechanism for the proxy specification. On the other hand, in these proposed schemes, the original signer is so powerful that he can forge proxy signatures. If the proxy signatures should be identified, this may causes critical damages on the benefit of the proxy signer. To overcome this unreasonable disadvantage, we propose an advanced proxy signature scheme. Then the advanced scheme provides the fair security protection for the proxy signers and the original signer.

Keyword: Proxy signature schemes, signature schemes.

1. INTRODUCTION

For the computer and communication security, the cryptography provides three basic services: secrecy, authenticity and integrity. The secrecy service is provided by cryptosystems while the other are provided by digital signature schemes. It is easy to see that the digital signature scheme plays an important role.

A digital signature is the electronic analog of a handwritten signature [5], so we hope that the application modes of digital signatures are similar to that of handwritten signatures. However, there exist some differences between digital signatures and handwritten signatures. Let us consider the following case. When someone goes on a vacation, he needs a proxy to execute his job. So the proxy is authorized to sign some kinds of documents on behalf of the original one during the vacation. But the digital signature scheme does not provide the proxy function.

In order to provide the proxy function, Mambo et al. proposed the concept of proxy signatures in [4]. In the proxy signature scheme, the original one is called the original signer while the proxy is called the proxy signer. The proxy signature scheme proposed in [4] allows the proxy signer to sign a message on behalf of the original signer. In their scheme, there are three kinds of proxy signatures: The full delegation, the partial delegation, and the delegation by warrant. Among these, the delegation by warrant is the most suitable one in security sense because the proxy signer is able to give the proxy specification in the warrant to protect his own right and benefit.

In [4], Mambo et al. pointed out seven properties of the proxy signatures: The unforgeability, the proxy signer's deviation, the secret-key dependence, the verifiability, the distinguishability, the identifiability, the undeniability. Due to the distinguishability, the verifier has to distinguish the valid proxy signatures generated by proxy from the signatures generated by the original signer. So, the original signer cannot forge proxy signatures for some proxy signer even if he has authorized the proxy signer. On the other hand, the signature receiver also needs to identify who is the signer. Therefore the secret key of the original signer and all of the proxy secret keys must be different.

In the past, many proxy signature schemes [1-4, 6, 8] were proposed. Among these, only the schemes in [1, 6] integrate the specification of the proxy period into the certificate of the proxy public key. Besides the proxy period, some other proxy specifications are also necessary and important. For example, it is necessary to specify what kinds of messages that the proxy signer could sign. In order to integrate all of the proxy specification into the proxy signature scheme, we will propose a new basic proxy signature scheme.

However, there exists another problem for these proxy signature schemes [1-4, 6, 8] because they do not satisfy the two important properties: the distinguishability and the identifiability. In other words, the original signer A can authorize himself to be a proxy signer B without the agreement of the user B. Then the original signer is able to forge the proxy signatures. No one can detect this illegal proxy case. The original signer may obtain some illegal benefits. To overcome this serious disadvantage, we will propose an advanced proxy signature scheme in this paper. Moreover, the advanced scheme satisfies the distinguishability and the identifiablity properties.

Since the basic scheme is based on Lee and Chang's group signature scheme [7], Lee and Chang's scheme is reviewed in Section 2. Then our schemes are described in Sections 3 and 4, respectively. In the same sections, the security analysis and discussions are also given. Finally, Section 5 is the conclusions.

2. REVIEW OF LEE AND CHANG'S SCHEME

Lee and Chang's group signature scheme is reviewed in this section. In their scheme, there are two public large prime numbers p and q such that q|p-1. In addition to p and q, there are a public one-way hash function h and a public generator g with order q in Z_q. A user A randomly selects his secret key $x_A \in Z^*_q$ and computes his public key $y_A = g^{x_A} \bmod p$. Here Z^*_q denotes the set of the positive integers which are smaller less than and relatively prime to q. Lee and Chang's scheme contains five phases: The certificate generation phase, the certificate key verification phase, the signature generation phase, the signature verification phase, and the disputation verification phase.

[The Certificate Generation Phase]

Suppose that the user A is a group owner and the user B wants to be a member of the group. After receiving the request from the user B, the owner A chooses a random integer $k \in Z^*_q$ and generates the certificate (U, V) as follows:

$$U = g^{-k} \times y_B^{k} \bmod p \text{ and}$$

$$V = k - U \times x_A \bmod q.$$

Then, he sends (U, V) to B.

[The Certificate Verification Phase]

After receiving (U, V), B verifies the certificate (U, V) by the verification equation

$$g^v \times (y_A)^U \times U \equiv (g^v \times (y_A)^U)^{x_B} \pmod{p}.$$

If the above equation holds, then the certificate (U, V) is valid. We explain why the verification equation holds in the following.

$$\begin{aligned} g^v \times (y_A)^U \times U \pmod{p} &\equiv DH_B \\ &\equiv g^{x_B \times k} \\ &\equiv (g^v \times (y_A)^U)^{x_B} \pmod{p} \end{aligned}$$

[The Signature Generation Phase]

Suppose that the member B wants to sign the message m. He first selects a random integer $t \in Z^*_q$ and computes $r= \alpha^t$ (mod p). Then he finds s such that $h(m) \equiv rx_B + ts$ (mod q). Then (r, s) is the signature of the message m. B sends (m, (r, s), (U, V)) to the verifier.

[The Signature Verification Phase]

In order to verify the signature (r, s), the verifier executes the following steps.

Step 1: Compute α by $\alpha= g^V \times (y_A)^U$ mod p.

Step 2: Compute DH_B by $DH_B= \alpha \times U$ mod p.

Step 3: Verify the signature (r, s) by the equation $\alpha^{h(m)} \equiv r^s \times (DH_B)^r$ (mod p). If above equation holds, the signature (r, s) is valid.

[The Disputation Verification Phase]

If the disputation occurs, with the help of the group owner A, then the verifier has the ability to find who is the signer.

Step 1: The group owner A chooses a random number a.

Step 2: The owner A computes r_A and s_A by $r_A= (g \times y_B)^a$ mod p and $s_A= a-r_A \times k$ mod q.

Step 3: The owner A sends (r_A, s_A) to the verifier and announces that the user B is the signer.

Step 4: The verifier computes $\beta_B= g \times y_B$ mod p and $\alpha= g^V \times (y_A)^U$ mod p.

Step 5: The verifier computes DH_B by $DH_B= \alpha \times U$ mod p.

Step 6: The verifier computes $\delta_B= \alpha \times DH_B$ mod p.

Step 7: The verifier checks whether or not the proxy signer is the user B by verifying $r_A \equiv \beta_B^{s_A} \times \delta_B^{r_A}$ (mod p).

3. THE BASIC PROXY SIGNATURE SCHEME

3.1. The Basic Scheme

In the basic scheme, p and q are two public large primes such that q is a prime factor of p-1. The public parameter g is a public generator with order q in Z_q and h is the public one-way hash function. The secret key of a user is a random integer $x \in Z^*_q$ and his public key is $y= g^x$ mod p. This basic scheme contains four phases: The certificate generation phase, the certificate verification phase, the signature generation phase, and the signature verification phase.

[The Certificate Generation Phase]

Suppose that a user A wants to assign a user B to be his proxy signer. Here the user A is called the original signer while the user B is called the proxy signer. In this phase, the proxy signer B could generate the proxy secret key x_p and the public key $y_p= g^{x_p}$ mod p. After receiving the public key y_p from B, A gives a certificate with a proxy warrant w to B. The proxy warrant w specifies the proxy period, the personal data of the proxy signer B, and the other important proxy data. To generate the certificate, the following procedure is performed.

Step 1: B selects a random integer $x_p \in Z^*_q$ as the proxy secret key and computes the public key $y_p= g^{x_p}$ mod p. B sends the public key y_p to the original signer A in a secure manner.

Step 2: After obtaining the public key y_p, the original signer A generates the certificate (U, V) for the warrant w as follows:

$$U= g^{-k} \times y_p^{\ k} \text{ mod p and}$$

$$h(w) \times V \equiv k-U \times x_A \text{ (mod q)},$$

where $k \in Z_q$ is a random integer and w is a proxy warrant. Then he sends (w, (U, V)) to B.

By the certificate (U, V) and the warrant w, the proxy signer B is able to show that he has the authorization from A during the valid proxy period.

[The Certificate Verification Phase]

After receiving (w, (U, V)) from the original signer A, B verifies (w, (U, V)) by the equation

$$(g^{V\times h(w)} \times y_A{}^U) \times U \equiv (g^{V\times h(w)} \times y_A{}^U)^{x_p} \pmod p.$$

If the above equation holds, then he gets the proxy authorization from the original signer A; otherwise, the proxy signer B has to request to produce the certificate again. Now the proxy signer B is able to sign some kinds of messages on behalf of the original signer A. The reason why Eq. (1) is correct is given as follows:

$$\begin{aligned}(g^{V\times h(w)} y_A{}^U) \times U &\equiv (g^{(k-U\times x_A)}\, g^{U\times x_A}) g^{-k} \times y_p{}^k \\ &\equiv y_p{}^k \\ &\equiv g^{k\times x_p} \\ &\equiv (g^{V\times h(w)} \times y_A{}^U)^{x_p} \pmod p.\end{aligned}$$

The above equation will hold if the certificate generated by the original signer A. Finally, the proxy signer B has the proxy secret key x_p and the proxy public key $y_p{}^k$ mod p.

[The Signature Generation Phase]

Suppose that the proxy signer B wants to sign a valid message m on behalf of A. He first selects a random number $t \in Z_q$ and computes r by the equation $r = (g^{V\times h(w)} y_A{}^U)^t$ mod p. Then he finds out s satisfying the equation $h(m) \equiv ts + rx_p \pmod q$. Then the proxy signature of the message m is (r, s). Finally, he sends ((m, (r, s)), (w, (U, V))) to the receiver.

[The Signature Verification Phase]

After receiving ((m, (r, s)), (w, (U, V))), the receiver first checks whether or not the proxy signer B still has the authorization to sign the message m. If the certificate (U, V) can not proved that the proxy signer has the authorization, the receiver rejects this proxy signature. Otherwise, the receiver verifies the proxy signature (r, s) by the following steps.

Step 1: Compute $g^k \equiv g^{V\times h(w)} \times y_A{}^U \pmod p$ and $y_p{}^k \equiv U \times (g^k) \pmod p$.

Step 2: Verify the signature (r, s) by $(g^k)^{h(m)} \equiv r^s \times (y_p{}^k)^r \pmod p$.

(1)

3.2 SECURITY ANALYSIS AND DISCUSSIONS

In this subsection, we analysis the security of our basic scheme. Let us consider the following possible attacks.

Attack 1: Acquire the proxy secret key x_p

There are three possible ways to compute the proxy secret key x_p.

(1) Acquire the proxy secret key x_p from the public key y_p. Because $y_p = g^{xp}$ mod p, then to acquire x_p from y_p is as hard as the discrete logarithm problem (DLP for short). So this way fails.

(2) Acquire the proxy secret key x_p from the proxy certificate (U, V). Though an attacker could compute $g^k \equiv g^{V\times h(w)} \times y_A{}^U \pmod p$ and the proxy public key $y_p{}^k \equiv U \times (g^{V\times h(w)} \times y_A{}^U) \equiv U \times (g^k) \pmod p$, it is also a DLP to find x_p from $y_p{}^k$ mod p.

(3) Acquire the proxy secret key x_p from the signature (r, s). Since s satisfies the equation $h(m) \equiv t \times s + r \times x_p \pmod q$, an attacker may try to find the secret key x_p from (r, s). However, in the above equation, there are two unknowns t and x_p, the attacker cannot determine the correct values of t and x_p.

Attack 2: Acquire the secret key x_A

Suppose that an attacker wants to compute the secret key x_A of the original signer A from the certificate (U, V). Since the equation $h(w) \times V \equiv k - U \times x_A \pmod q$ contains two unknown variables k and x_A, the attacker cannot get the correct secret key x_A.

Attack 3: Acquire the secret random integer k

It is hard to find k from $g^k \equiv g^{V\times h(w)}\times y_A{}^U$ (mod p) because this is a DLP. The attacker cannot determine the value of k by the equation $h(w)\times V\equiv k- U\times x_A$ (mod q) since the two variables k and x_A are unknown.

Attack 4: Acquire the secret random number t

Because the equation $h(m)\equiv ts+ rx_p$ (mod q), if the attacker knows x_p, then he is able to determine the correct value of t. Fortunately, x_p is secret, so this attack finding t fails. On the other hand, the attacker cannot compute t from r since $r=(g^k)^t$ mod p.

Attack 5: Forge the certificate (U, V) without the secret key x_A.

If the forger generates (U, V) by first determining the value of k, then he easily computes U but hardly to executes $h(w) \times V\equiv k- U\times x_A$ (mod q). The reason is that he does not have the secret key x_A. On the other hand, if he gives V first, then he needs to solve the equation $(g^{V\times h(w)}\times y_A{}^U)\times U\equiv y_p{}^{V\times h(w)}\times(y_A{}^U)^{xp}$ (mod p). However, the difficulty of this problem is similar to that of the DLP even if the attacker has x_p. Because (U, V) cannot be forged, the integrity of the warrant w is also guaranteed by (U, V).

Attack 6: Forge a valid proxy signature (r, s) without the secret key x_p.

Due to the analysis of Attack 5, no one can forge the certificate (U, V). So the proxy public key $y_p{}^k$ mod p is fixed and authenticated. Similarly, without the secret key x_p, the forger cannot determine the value of s by first determining t and r. If the forger determines s first, he cannot determine the value of r satisfying the equation $r^s\equiv (g^k)^{h(m)}/ (y_p{}^k)^r$ (mod P).

In our basic scheme, the original signer can control the proxy period and proxy right through the certificate (U, V) and the warrant w. The proxy period and proxy right are specified in the proxy warrant w. The integrity of the warrant w is guaranteed by the certificate (U, V). Therefore, any proxy signer cannot generate the proxy signatures that are not authorized in the proxy warrant w. Moreover, this basic scheme provides the anonymity for the proxy signers if the identification of the proxy signer is not necessary.

However, if the original signer needs to point out who is the proxy signer, there exists a serious problem. The basic assumption of the basic scheme is that the original signer is trustworthy. If the original signer is not trustworthy, he could pretend that he is a proxy signer. He is easy to do that because he could execute all of the steps in the certificate generation and verification phases without the agreement of the proxy signer. No one could find this case. It is unreasonable for proxy signers. To overcome this disadvantage, we will propose an advanced scheme in the next section.

4. THE ADVANCED PROXY SIGNATURE SCHEME AND DESCRIPTION

4.1 The Advanced Scheme

Now we propose an advanced proxy signature scheme in which the original signer cannot pretend that he is a proxy signer. At the same time, the receiver of proxy signatures could identity the proxy signer. Initially, there are two public large primes p and q such that q|p-1. The parameter g is the generator with order q in Z_p and α is the primitive element in Z_q. In addition, h is a public one-way hash function. Here both of the discrete logarithm problems in Z_p and in Z_q should be infeasible. Each user selects a random integer $x\in Z^*_q$ as his secret key. Then the user has two public keys: $y= g^x$ mod p and $y'= \alpha^x$ mod q. This new scheme contains four phases: The certificate generation phase, the certificate verification phase, the signature generation phase, and the signature verification phase.

[The Certificate Generation Phase]

Let the users A and B be the original signer and the proxy signer, respectively. In this phase, the original signer A creates a certificate with the warrant w for the proxy signer B. The original signer A first selects a random integer $k \in Z^*_q$, computes the certificate (U, V) as follows:

$$U= \alpha^k \bmod q \text{ and}$$
$$V= h(w) \times x_A + U \times k \pmod{(q-1)}.$$

Then the original signer A gives (w, (U, V)) to the proxy signer B.

[The Certificate Verification Phase]

The proxy signer B uses the equation $\alpha^V \equiv (y'_A)^{h(w)} \times U^U$ (mod q) to verify the certificate (U, V). If the equation does not hold, then B requests A to produce the certificate again.

[The Signature Generation Phase]

Suppose that B wants to sign the message m. First, he chooses a random integer $t \in Z^*_q$. He generates the signature (r, s) by computing $r= g^t \bmod p$ and finding s such that $h(m) \equiv t \times s + r \times U \times x_B \pmod q$. Then he sends ((m, (r, s)), (w, (U, V))) to the verifier.

[The Signature Verification Phase]

In this phase, the verifier has to first check the validity of the certificate (U, V) with warrant w. If the certificate (U, V) and the warrant w are valid, then he verifies the proxy signature (r, s).

Step 1: Check whether or not the proxy signer has the authorization to sign the message m according to the warrant w. If the authorization is not valid, then the verifier rejects the signature (r, s).

Step 2: Verify (U, V) by the equation $\alpha^V \equiv (y_A')^{h(w)} \times U^U \pmod q$. If the certificate (U, V) is not valid for the warrant w, the verifier has to reject the signature (r, s).

Step 3: Check the signature (r, s) by the equation $g^{h(w)} \equiv r^s \times (y_B)^{r \times U} \pmod p$.

4.2 SECURITY ANALYSIS AND DISCUSSIONS

Let us consider the following possible attacks for this advanced scheme.

Attack 1: Acquire the secret key x_A

There may be two ways to acquire the secret key x_A.

(1) Acquire the secret key x_A from the public keys y_A or y_A'. Because $y_A= g^{x_A} \bmod p$ and $y_A'= \alpha^{x_A} \bmod q$, the difficulty acquiring x_A from the public keys y_A or y_A' is the same as that of the discrete logarithm problem. So the public keys y_A and y_A' do not reveal the secret key x_A.

(2) Acquire the secret key x_A from the certificate (U, V). Due to the equation $V= h(w)x_A + U \times k \pmod{(q-1)}$ has two unknown parameters k and x_A, the attacker cannot obtain x_A from the certificate (U, V).

Attack 2: Acquire the secret key x_B

There are also two possible ways to acquire the secret key x_B.

(1) Acquire the secret key x_B from the public keys y_B or y_B'. Because $y_B= g^{x_B} \bmod p$ and $y_B'= \alpha^{x_B} \bmod p$, it is as difficult as the discrete logarithm problem. So it is hard to compute x_B from the public keys y_B or y_B'.

(2) Acquire the secret key x_B from the signature (r, s). The attacker needs to solve the equation $h(m) \equiv t \times s + r \times U \times x_B \pmod q$. However, the equation contains two unknown parameters, t and x_B, he cannot determine the value of x_B.

Attack 3: Acquire the secret random integer k

To compute the integer k from U is as hard as to solve

DLP for U= α^k mod q. On the other hand, it also impossible to obtain the integer k from the equation V≡ h(w)x_A+U×k (mod (q-1)) without the secret key x_A.

Attack 4: Acquire the secret random integer t

Since r= g^t mod p, then it is the DLP to obtain t from r. To obtain t from the equation h(m)≡ t×s+ r×U×x_B (mod q) cannot work because the attack does not have the secret key x_B.

Attack 5: Forge the certificate (U, V).

In order to forge the certificate (U , V) for the warrant w', the attacker may first selects a random integer k . Then he easily computes U =$\alpha^{k'}$ mod q. But it is hard for him to find V' such that V = h(w)x_A+U ×k (mod (q-1)) without the secret key x_A. Only the user A can generate the certificate (U, V). In other words, there is no one can forge valid proxy signatures without the authorization of the user A.

Attack 6: Forge the proxy signature (r, s).

To forge the proxy signature (r', s'), the attacker may first give the value of t and compute r'. Without the secret key x_B, it is hard to find s such that h(m)≡ t ×s +r'×U×x_B (mod q). The original signer A also cannot generate the proxy signature (r', s') without the secret key x_B even if he is able to construct the valid certificate (U, V). Therefore, the users, including the original signer A, cannot forge the proxy signatures.

Due to the above analysis, the new scheme also provides the warrant w in which the important original signer can clearly describe the proxy period, the identity of proxy signer and the other important proxy data. The warrant w can be authenticated by the certificate (U, V). Then the proxy signer cannot sign any message without the authorization of the original signer.

On the other hand, the right of the proxy signer is also protected by this advanced scheme. In this scheme, the original signer cannot forge the proxy signatures. Moreover, the verifier can verify who is the real signer and check whether or not the signatures are proxy signatures. Therefore, the advanced scheme provides the fair protection for the original signer and the proxy signers.

5. CONCLUSIONS

In this paper, two specifiable proxy signature schemes are proposed. Since 1996, many proxy signature schemes were proposed. Two of the proposed schemes provide the specification of the proxy period [1, 6]. However, the proxy period is not the only important proxy information. To integrate the complete specification of the proxy data into the proxy signature scheme, our basic scheme is proposed. In the basic scheme, the warrant is able to describe the complete proxy specification that the original signer wants. To guarantee the integrity and authentication of the warrant, the original signer generates the certificate. Without the certificate, no one can generate proxy signatures without the authorization of the proxy signer. Using the certificate and the warrant, the proxy signer shows that he is able to sign some authorized messages on behalf of the original signer. With the help of the warrant and the certificate, the basic scheme provides the complete specification function. If the original signer does not need to point out the identity of the proxy signer, the basic scheme could also provide the anonymity protection for the proxy signers.

The basic security assumption of the basic scheme is that the original signer must be trustworthy. If the original one is not trustworthy, in the basic scheme, the original signer is so powerful that he can pretend that he is some proxy signer. This illegal case is undetectable. To overcome this problem, we propose the advanced scheme. In the advanced scheme, though the original signer is able to generate the certificate, he cannot forge proxy signatures since he does not have the secret key of the proxy signer. So the receiver of the proxy signatures can verify who is the real signer. At the same time, the receiver can check whether or not the signatures are proxy signatures. Our advanced scheme satisfies the distinguishability and the indetifiiablity properties given in [4] while the other proposed schemes do not satisfy these two properties. Moreover, the advanced scheme provides fair protection for the proxy signers and the original signer.

References

[1] Sun, Hung-Min and Chen, Biing Jang: "Time-Stamp Proxy Signatures with Traceable Receivers," Proceedings of the Ninth National Conference on Information Security, Taiwan, 1999, pp. 247-253.

[2] *Sun Hung Min, and Hsieh, Bin Tsan*: "Remark on Two Nonrepudiable Proxy Signature Schemes," Proceedings of the Ninth National Conference on Information Security,Taiwan, 1999, pp. 241-246.

[3] NAMBO Masahiro, USUDA Keisuke, and OKAMOTO Eiji: "Proxy Signatures: Delegation of the Power to Sign Message," ICICE. Trans. Fundamentals, E 79-A, 9, 1996, pp.1338-1354.

[4] NAMBO, Masahiro, USUDA, Keisuke, and OKAMOTO, Eiji: "Proxy Signatures for Delegation Signing Operation," Proc. 3rd ACM Conference on Computer and Communications Security, 1996, pp. 48-57.

[5] Nechvatal and James: "Public Key Cryptography," in Contemporary Cryptology: The Science of Information Integrity, Simmons, G. J. ed., IEEE Press, Piscataway, N. J, 1991, pp. 177-288.

[6] Lee, Narn-Yih, Hwang, Tzonelih, and Wang, Chih Hung: "On Zhang's Nonrepudiable Proxy Signature Schemes," Third Australasian Conference, ACISP '98, 1998, pp. 415-422.

[7] Lee, Wei Bin and Chang, Chin Chen: "Efficient Group Signature Scheme Based on the Discrete Logarithm," IEE Proc.-Comput. Digit Tech., Vol. 145, No. 1, Jan. 1998, pp. 15-18.

[8] Zhang K: "Threshold proxy signature schemes," 1997 Information Security Workshop, Japan, Sep. 1997, pp. 191-197.