可轉換之代理簽章系統
Convertible Proxy Signature Scheme

Hung-Min Sun(孫宏民)
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan 701
Email: hmsun@mail.ncku.edu.tw

Abstract

In this paper, we propose a new type of proxy signature, called convertible proxy signature scheme. A convertible proxy signature scheme allows a proxy signer to sign a message on behalf of an original signer. The proxy signer can create a valid proxy signature for the original signer. However, the identity of the proxy signer isn't exposed to any other third party from the proxy signature. To avoid a dispute, the proxy signer must have the capability of proving, to any arbitrator, that he is the actual proxy signer of a valid proxy signature. On the contrary, a proxy signer must also have the capability of proving, to any arbitrator, that he isn't the actual proxy signer of a valid proxy signature which is created by others. In case of necessity, the original signer or the proxy signer can convert the proxy signature into a new proxy signature in which the identity of the proxy signer is disclosed. Based on the discrete logarithm problem, a concrete convertible proxy signature scheme is realized in this paper.

Keywords: *Cryptography, Digital Signature, Proxy Signature, Discrete Logarithm*

1 Introduction

The concept of delegation has been proposed in various applications, such as, delegation to sign messages [1-2], delegation to decrypt messages [3], delegation to give a vote [4]. The proxy signature scheme [1-2] was proposed to realize the concept of delegation to sign messages. A proxy signature scheme allows a designed person, called a proxy signer, to sign messages on behalf of an original signer. So far, there have been three types of delegation -- *full delegation*, *partial delegation*, and *delegation by warrant*. Under full delegation, a proxy signer is given the same secret that the original signer has. Therefore, a proxy signer can create the same signature as the original signer does. Under partial delegation [1-2,5], the original signer uses the original signature key to create a proxy key, which will then be sent to the proxy signer. For security requirements, the original signature key should not be computed from the proxy key. The proxy signer can use the proxy key to sign messages on behalf of the original signer. Under delegation by warrant, a warrant, which certifies the proxy signer, is issued to the proxy signer from the original signer. With the warrant, the signatures created by the proxy signer are regarded to have the same validity as those signed created by the original signer. This type of delegation has appeared in [6-7].

There were many proxy signature schemes constructed for each of these delegation types in the past [1-2,5-10]. The partial delegation and delegation by warrant are more secure than the full delegation. Compared with the delegation by warrant, the partial delegation has faster processing speed. Therefore, among these three types of delegation, the partial delegation is the best choice to use. For simplicity, in this paper we call the partial delegation the proxy signature if it doesn't lead to any confusion.

Among those existing proxy signature schemes [1-2,8-10], either the proxy signer is anonymous to any third party [1-2], or the identity of the proxy signer is exposed from the proxy signature [1,8-10]. Proxy signature schemes with the property of hiding the identity of the proxy signer to any other third party are very important in some special environments. For example, in a secret organization, it is usually expected not to expose the identities, tasks, and positions of their members, due to some security reasons. In the past, a proxy signature scheme that satisfies the above property was proposed by Mambo *et al.* [1-2]. However, the proposed scheme has the drawback that the original signer must be trustworthy enough and never cheat. Or, the original signer can cheat any verifier by creating an acceptable proxy signature. This is because the proxy signer has no way to show his identity to the verifier (or an arbitrator). Proxy signature schemes with the property of disclosing the identity of the proxy signer from a proxy signature are also very important in various applications. This is usually for the responsibility of the proxy signer. Once a proxy signature is created by a proxy signer, it should not be disavowed by the proxy signer. In the past, some proxy-protected proxy signature schemes (or called non-repudiable proxy signature schemes) [1,8-10] were proposed to achieve the above property. In a proxy-protected proxy signature scheme, a verifier can determine the identity of the proxy signer from a proxy signature, while the proxy signer cannot repudiate the creation of a valid signature against anyone later. Thus the drawback in the proxy signature scheme with hiding the identity of the proxy signer can be overcame. But, on the contrary, these proxy-protected proxy signature schemes lose the ability to protect the identity of the proxy signer. Therefore, we desire a proxy signature scheme having these two advantages simultaneously.

In this paper, we propose a new type of proxy signature, called convertible proxy signature scheme. A convertible proxy signature scheme allows a proxy signer to sign a message on behalf of an original signer. The proxy signer can create a valid proxy signature for the original signer. However, the identity of the proxy signer isn't exposed to any other third party from the proxy signature. To avoid a dispute, the proxy signer must have the capability of proving, to any arbitrator, that he is the actual proxy signer of a valid proxy signature. On the contrary, a proxy signer must also have the capability of proving, to any arbitrator, that he isn't the actual proxy signer of a valid proxy signature which is created by others. In case of necessity, the original signer or the proxy signer can convert the proxy signature into a new proxy signature in which the identity of the proxy signer is disclosed. In this paper, based on the discrete logarithm problem, we give a concrete realization of convertible proxy signature scheme. This paper is organized as follows. In section 2, we describe the requirements for a convertible proxy signature scheme. In section 3, we propose a convertible proxy signature scheme based on the discrete logarithm problem. Security considerations of the proposed scheme are analyzed in section 4. Finally, we conclude this paper in section 5.

2 Requirements for a convertible proxy signature scheme

A convertible proxy signature scheme should satisfy several requirements. Here we describe these requirements in the following. Note that part of these requirements come from the requirements for proxy signatures, proposed by Mambo et al. [1-2].

(1) **Unforgeability:** The proxy signer can create a valid proxy signature on behalf of the original signer. Any other third party, except the original signer, is unable to forge a valid proxy signature.

(2) **Verifiability:** Any verifier can check the validity of a proxy signature.

(3) **Identifiability:** The original signer can determine the identity of the actual proxy signer from a proxy signature.

(4) **Anonymity:** The actual proxy signer of a proxy signature is anonymous to any other third party. That is, the identity of the proxy signer isn't exposed to any other third party from the proxy signature.

(5) **Confirmation:** A proxy signer can prove, to any arbitrator, that he is the actual proxy signer of a valid proxy signature which is created by him. As mentioned in Section 1, it is possible that the original signer creates an acceptable proxy signature and then claims that the proxy signature is created by the proxy signer. Therefore, in case of a dispute, we need an arbitrator to judge who is the cheater. Note that no other task is assigned to the arbitrator in addition to judgement. Therefore, he is unable to convert the proxy signature into a new proxy signature with the property of disclosing the identity of the proxy signer.

(6) **Disavowal:** A proxy signer can prove, to any arbitrator, that he isn't the actual proxy signer of a valid proxy signature which is created by others.

(7) **Convertibility:** By releasing some secret parameters from the original signer or the actual proxy signer, a proxy signature can be converted into a new proxy signature with the property of disclosing the identity of the actual proxy signer. The converted proxy signature cannot be forged by anyone (including the original signer), and can be verified by any verifier. For the converted proxy signature, the actual proxy signer cannot disavow it.

3 The Proposed Scheme

In the following section, we propose a scheme to realize the concept of the convertible proxy signature scheme. The proposed scheme is based on the discrete logarithm problem.

Initialization:

Let p be a large prime, q be a prime factor q of p-1, and g be an element of order q in Z_p^*. The original signer, Alice, has a private key $x_{Alice} \in Z_q$ and a public key $y_{Alice} = g^{x_{Alice}}$ (mod p). The proxy signer, Bob, has a private key $x_{Bob} \in Z_q$ and a public key $y_{Bob} = g^{x_{Bob}}$ (mod p). Both y_{Alice} and y_{Bob} are certified by a certification authority (CA). $h(\,)$ is a public one-way hash function [11].

Proxy key generation phase:

Step 1. Alice selects two random numbers r_1, $r_2 \in Z_q$, and computes $K_1 = (y_{Bob})^{r_1}$ (mod p), $K_2 = g^{r_2}$ (mod p), and $K_3 = g^{r_1}$ (mod p). Let W_{Alice} record the information describing the original signer's identity, the valid delegation time, and other information on the delegation for the security requirements. Note that the proxy signer's identity is not included in W_{Alice}. Alice concatenates W_{Alice}, K_1, K_2, K_3, and hashes the result:

$$e = h(W_{Alice}, K_1, K_2, K_3).$$

Step 2. Alice computes $\sigma = e\,x_{Alice} + r_2$ (mod q), and sends (σ, W_{Alice}, r_1, K_2) to Bob in a secure manner.

Step 3. Bob computes $K_1 = (y_{Bob})^{r_1}$ (mod p), $K_3 = g^{r_1}$ (mod p), and $e = h(W_{Alice}, K_1, K_2, K_3)$. Then he verifies the validity of σ by checking if the following equation holds:

$$g^{\sigma} = (y_{Alice})^{e}\, K_2.$$

Step 4. Bob computes $\sigma_p = \sigma + r_1 \cdot x_{Bob}$ (mod q) and accepts σ_p as a valid proxy key. Hence,

$$g^{\sigma_p} = y_{Alice}{}^{h(W_{Alice}, K_1, K_2, K_3)} K_1 K_2 \pmod{p}.$$

Note that in Step 1, the parameters K_1 and r_1 need to be kept by the original signer. These parameters will be used to identify the proxy signer and to convert the proxy signature later.

In Step 4, it is difficult for a forger to find σ_p', K_1', K_2', and K_3' such that the following equation holds: $g^{\sigma_p'} = y_{Alice}{}^{h(W_{Alice}, K_1', K_2', K_3')} K_1' K_2'$ (mod p). This is because he need solve the discrete logarithm problem. The detailed cryptanalysis on the proposed scheme is given in Section 4.

Proxy signature generation and verification phase:

Step 5. Bob uses the proxy key σ_p to sign a message m and generate a signature S_m by using a conventional digital signature scheme [12-14]. Then the proxy signature is (m, S_m, W_{Alice},

K_1, K_2, K_3).

Step 6. Any verifier can check the validity of the proxy signature (m, S_m, W_{Alice}, K_1, K_2, K_3) by using a new public key y', where $y' = y_{Alice}^{h(W_{Alice},K_1,K_2,K_3)} K_1 K_2 \pmod{p}$, in the signature verification phase of the conventional digital signature scheme.

It is clear that the identity of the proxy signer is not exposed from the proxy signature or the signature verification process. Therefore, the proxy signer is anonymous to any other third party.

In some applications, it is sometimes necessary for an arbitrator to verify if a proxy signer is the actual proxy signer of a valid proxy signature. In the following, we show that our scheme has the capability to can verify whether a proxy signer is the actual proxy signer of a valid proxy signature or not.

Signer confirmation phase:

Bob can convince any arbitrator that a valid proxy signature is indeed generated by him. This is because he can prove that the discrete logarithm of K_1 to the base K_3 is the same as the discrete logarithm y_{Bob} to the base g using the zero-knowledge protocol of the equality of two discrete logarithms due to Chaum [15].

Signer disavowal phase:

Bob can convince any arbitrator that a proxy signature generated by others is not indeed generated by him. This is because he can prove that the discrete logarithm of K_1 to the base K_3 is different from the discrete logarithm y_{Bob} to the base g using the zero-knowledge protocol of the inequality of two discrete logarithms due to Chaum [15].

Converting proxy signature phase:

In our scheme, the original signer or the proxy signer can release the secret parameter r_1 to convert a valid proxy signature into a proxy signature with the known proxy signer. Let ID_{Bob} denote the identity of the proxy signer. Because $K_1 = y_{Bob}^{r_1} \pmod{p}$ and $K_3 = g^{r_1} \pmod{p}$, the converted proxy signature is (m, S_m, W_{Alice}, r_1, K_2, ID_{Bob}). The validity of the converted proxy signature can be verified by any verifier using the public key y', where $y' = y_{Alice}^{h(W_{Alice}, y_{Bob}^{r_1}, K_2, g^{r_1})} y_{Bob}^{r_1} K_2 \pmod{p}$, in the signature verification process that the conventional digital signature scheme used. Here y_{Alice} and y_{Bob} are the public keys of the original signer and proxy signer. Note that the information ID_{Bob} in the converted proxy signature is only used for a verifier to get the public y_{Bob} from the CA. It is difficult for a forger Carol to cheat by replacing ID_{Bob} with her identity ID_{Carol} because the following equation doesn't hold: $y' = y_{Alice}^{h(W_{Alice}, y_{Carol}^{r_1}, K_2, g^{r_1})} y_{Carol}^{r_1} K_2 \pmod{p}$.

4 Security Considerations

In this section, we consider the security of the proposed scheme by examining if the proposed scheme achieves the requirements for a convertible proxy signature scheme, as mentioned in Section 2.

On unforgeability:

Because σ_p is created in the proxy key generation phase, in which only the original signer and the proxy signer participate, a forger is unable to obtain σ_p to sign messages instead of the proxy signer in the proxy signature generation phase. If a forger attempts to forge a valid proxy signature for the original signer, he must find σ_p', K_1', K_2', and K_3' such that the following equation holds: $g^{\sigma_p'} = y_{Alice}^{h(W_{Alice},K_1',K_2',K_3')} K_1' K_2' \pmod{p}$. If K_1', K_2', and K_3' are predetermined, he must solve the discrete logarithm problem, which is infeasible, to obtain σ_p'. If σ_p', K_1', and K_2' are predetermined, he must solve the discrete logarithm problem to obtain the value $h(W_{Aliece}, K_1', K_2', K_3')$ and invert the hash function to obtain K_3', which is more difficult than the previous case. If σ_p', K_1', and K_3' (or σ_p', K_2', and K_3') are predetermined, it is difficult to compute K_2' (or K_1') because K_2' (or K_1') appears twice in the verification equation and a hash function is applied to it.

On verifiability:

As described in Step 5, any verifier can check the validity of the proxy signature.

On identifiability:

We assume that there are multiple proxy signers for an original signer. Because the parameters K_1 and r_1 have been kept by the original signer, the original signer can identify the proxy signer from the proxy signature by checking if $K_1 = (y_{Bob})^{r_1} \pmod{p}$.

On anonymity:

Given a proxy signature, the only information which can be used to identify the proxy signer is to check if $K_1 = (y_{Bob})^{r_1} \pmod{p}$. However, the value r_1 is unknown by any other third party. Thus, the identity of the proxy signer isn't exposed to any other third party from a proxy signature.

On confirmation:

As described in the signer confirmation phase, the proxy signer can prove, to any arbitrator, that he is the proxy signer of a valid proxy signature.

On disavowal:

As described in the signer disavowal phase, a proxy signer can prove, to any arbitrator, that he isn't the proxy signer of a valid proxy signature which is not created by him.

On convertibility:

As described in the converting proxy signature phase, a proxy signature can be converted into a new proxy signature with the property of disclosing the identity of the proxy signer. Basically, the security analysis on unforgeability for the converted proxy signature is the same as in that of the original proxy signature. Here we'd like to include more security analysis on whether the original signer can forge a valid converted proxy signature. From Step 4, we know that $\sigma_p = \sigma + r_1 \cdot x_{Bob}$. Among these parameters σ_p, σ, r_1, and x_{Bob}, the original signer knows only σ and r_1. The parameters x_{Bob} is the proxy signer's private key which is unknown by the original signer. Therefore, the original signer cannot obtain σ_p.

5 Conclusions

In this paper, we present a new concept of convertible proxy signature scheme in which the actual proxy signer is anonymous to any other third party initially. In case of a dispute, the actual proxy signer is able to prove, to any arbitrator, that he is the actual proxy signer of a valid proxy signature. In addition, a proxy signer can prove to any arbitrator that he isn't the actual proxy signer of a valid proxy signature which is created by others. Thus, no one, including the original signer, can pretend the actual proxy signer. Also, the actual proxy signer cannot disavow that he is the actual proxy signer. In case of necessity, the proxy signature can also be converted into a proxy signature with the property of disclosing the identity of the proxy signer. Thus the proxy signer cannot deny, to any verifier, what he has ever signed.

References

[1] MAMBO, M., USUDA, K., and OKAMOTO, E.: 'Proxy signatures: Delegation of the power to sign messages'. *IEICE Trans. Fundamentals*, 1996, E79-A, 9, pp. 1338-1354

[2] MAMBO, M., USUDA, K., and OKAMOTO, E.: 'Proxy signatures for delegating signing operation'. *Proc. 3rd ACM Conference on Computer and Communications Security*, (ACM press, 1996), pp. 48-57

[3] MAMBO, M. and OKAMOTO, E.: 'Proxy cryptosystems: Delegation of the power to decrypt ciphertexts'. *IEICE Trans. Fundamentals*, 1997, E80-A, 1

[4] PETERSEN, H. and Horster, P.: 'Self-certified keys - concepts and applications'. *Proc. 3rd Conference on Communications and Multimedia Security*, Sep. 22-23, 1997

[5] Usuda, K., MAMBO, M., UYEMATSU, T., and OKAMOTO, E.: 'Proposal of an automatic signature scheme using a complier'. : *IEICE Trans. Fundamentals*, 1996, E79-A, 1, pp. 94-101

[6] VARADHARAJAN, V., ALLEN, P., and BLACK, S.: 'An analysis of the proxy problem in distributed systems'. *Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, (1991), pp. 255-275

[7] NEUMAN, B.C., 'Proxy-based authorization and accounting for distributed systems'. *Proc. 13th International Conference on Distributed Systems*, 1993, pp. 283-29157

[8] KIM, S., PARK, S., and WON, D.: 'Proxy signatures, revisited'. *ICICS'97, Lecture Notes in Computer Science* 1334, (Springer-Verlag, 1997) pp. 223-232

[9] ZHANG, K.: 'Threshold proxy signature schemes'. *1997 Information Security Workshop*, Japan, Sep., 1997, pp. 191-197.

[10] LEE, N.Y., HWANG, T., and WANG, C.H.: 'On Zhang's nonrepudiable proxy signature schemes' ACISP'98, *Lecture Notes in Computer Science* 1438, (Springer-Verlag, 1998), pp. 415-422

[11] RIVEST, R.L.: 'The MD5 message digest algorithm'. RFC 1321, Apr 1992.

[12] ELGAMAL, T.: 'A public key cryptosystem and signature scheme based on discrete logarithms'. *IEEE Tran. Information Theory*, 1985, 31, (4), pp. 469-472

[13] NYBERG, K. and RUEPPEL, R.A.: 'A new signature scheme based on the DSA giving message recovery'. *Proc. 1st ACM conference on Computer and Communications Security*, 1993.

[14] HORSTER, P., MICHELS, M., and PETERSEN, H.: 'Meta-ElGamal signature schemes'. *Proc. 2nd* ACM *Conference on Computer and Communications Security*, (ACM press, 1994), pp. 96-107.

[15] CHAUM, D.: 'Zero-knowledge undeniable signatures'. *Lecture Notes in Computer Science 473, Advances in Cryptology –Eurocrypt'90 Proceedings*, (Springer-Verlag, 1991), pp. 458-464.