A Proxy Signature Scheme without Using One-way Hash Functions

*Hwang, Shin-Jia, and Shi, Chi-Hwai**
Department of Information Management, Chaoyang University of Technology
Wufeng, Taichung Country, 413, Taiwan, R.O.C.
*Department of Computer Science, National Chung Hsing University
250, Kuo Kuang Road, Taichung, 402, Taiwan, R.O.C.
Email: sjhwang@cyut.edu.tw
*Email: chshi@cs.nchu.edu.tw

ABSTRACT

A new proxy signature scheme without using one-way hash functions is proposed. This scheme has the following advantages. This new scheme does not need one-way hash functions to integrate warrants into the proxy certificate. Therefore, the overall security of this proxy signature scheme is only based on the discrete logarithm problem. This new scheme also removes the requirement of secure channels while the most proposed proxy signature schemes need. Our scheme provides the powerful specifiable function for the proxy details with the help of warrants. Moreover, the new scheme supports a fair protection both for the original and proxy signers.

KEYWORDS: Proxy signatures, digital signatures, one-way hash functions

1. INTRODUCTION

Digital signature schemes provide two important cryptographic functions: Integrity and authenticity [7]. However, it is not enough to solve some practical applications only by digital signature schemes. For example, some user wants to take a vacation, so he needs a proxy agent to execute his job. The proxy agent should be able to sign some documents on behalf of the user. But digital signature schemes do not provide the proxy function by which the user is able to authorize someone as his proxy agent.

In order to provide the proxy function, Membo et al. [5, 6] proposed the concept of the proxy signature schemes in 1996. In a proxy signature scheme, a user, an original signer, could designate someone as his proxy signer to sign a message on behalf of him. Since then, many proxy signature schemes [3-6, 8-11, 13] were proposed. Among these proxy signature schemes, there are three kinds of delegation ways: The full delegation, the partial delegation, and the delegation by warrants. In the full delegation, the original signer shares his secrete key with the proxy signer. But the full delegation causes some authentication problem for the release of the secret key of the original signer. The authentication problem means that the verifier cannot identify the actual signer of signatures.

To solve the authentication problem, both in the partial delegation and the delegation by warrants, a proxy secret key and a proxy public key are generated. To guarantee the authorization of the proxy secret key and proxy public key, the original signer generates a proxy certificate for the proxy public key by his secret key. Then any user is able to check whether the proxy signer has the proxy authorization.

The delegation by warrants is more specifiable than the partial delegation. In the delegation by warrants, a warrant is used to describe all proxy details such as the proxy period, the names of the proxy and original signers, and the responsibility of the proxy signer. On the other hand, the partial delegation does not have the function to describe the proxy details. Moreover, warrants can also prevent that the proxy signer transfers any legal proxy authorization to another user without the agreement of the original singer.

In the delegation by warrants, to integrate the warrants into the proxy certificate, the one-way hash function is used. The use of one-way hash functions may weaken the overall security of the proxy signature scheme. The security of digital signature schemes is based on some famous cryptographic problems, such as the discrete logarithm problem, while the security of most one-way hash functions is not [1]. Moreover, in general, the lifetime of the one-way functions is shorter than that of digital signature scheme.

Instead of the security of a proxy signature scheme based on the weaker assumption between the signature scheme and the one-way hash function, we propose a proxy signature scheme without one-way hash functions. In the next section, the new scheme is described. The security analysis and discussions are given in Section 3. Section 4 is our conclusions.

2. OUR NEW SCHEME WITHOUT ONE-WAY HASH FUNCTIONS

There are two public large prime numbers P and Q such that Q is a large prime factor of P-1. The public parameter g is a generator with order Q in Z_P. There is another public large prime number P' such that P'> Q. The public parameter α is a primitive root in $Z_{P'}$. The each user i randomly selects his secret key $x_i \in Z_Q$ and computes his public key $y_i= g^{x_i}$ mod P.

Suppose that the original signer A wants to authorize the user B as his proxy signer. The specification of the proxy is described in a warrant w. The warrant w is a short document in some special formats. Some of the special formats come of the data with predetermined formats such as the personal identities, the user name and address. Moreover, the warrant will be usually written to a set format. On the other hand, the length of a warrant w may be short because the warrant w only records some necessary proxy details.

The new scheme contains two phases: The proxy key generation phase and the signature generation and verification phase. In the following, we describe the two phases, respectively.

[The proxy key generation phase]

Step 1: The original signer A chooses a random integer $k' \in Z_Q$ and computes $r'= g^{k'}$ mod P. He also compute W= (α^w mod P') mod Q. Then A sends (w, r') to the proxy signer B.

Step 2: The proxy signer B selects a random integer $a \in Z_Q$ and computes $r= g^a r'$

mod P and r''= $(y_A)^a$ mod P. B also compute W= (α^w mod P') mod Q. Then he sends r'' to A.

Step 3: The original signer A computes r= $(r'')^{x_A^{-1}}$ r' mod P and s'= k'+ Wrx_A mod Q and sends s' to B.

Step 4: The proxy signer verifies s' by the equation $g^{s'} \equiv r'(y_A)^{rW}$ (mod P). If the equation holds, the proxy signer has obtains the authorization form the original signer A. Then the proxy signer computes the proxy secret key s= s'+a+rx_B mod Q and the proxy public key g^s mod P.

[The signature generation and verification phase]

Assume that the proxy signer B wants to sign a message m on behalf of the original signer A. By using the proxy secret key s and the digital signature schemes based on the discrete logarithm problem, the proxy signer B is able to generate the signature $Sign_s(m)$. Then he sends ((w, r), (m, $Sign_s(m)$)) to the verifier.

According to the specification in the warrant w, the verifier first checks whether or not the user B has the authorization to sign the message m. For the proxy signer B, the verifier recovers the proxy public key by the equation $g^s \equiv r(y_A)^{rW} y_B{}^r$ mod P, where W= (α^w mod P') mod Q. Then he use this proxy public key to verify whether or not the signature $Sign_s(m)$ is signed by the proxy signer B.

3. SECURITY ANALYSIS AND DISCUSSIONS

In essence, the procedure to generate (w, r) is a digital signature scheme based on the discrete logarithm problem. Given a warrant w, no one can generate (r, s') without the secret key of the original signer. Given a (r, s'), the proxy signer may try to the attack which the both sides of s'= k'+ Wrx_A mod Q are multiplied by a factor α. However, he cannot determine the value of w from W because W= (α^w mod P') mod Q is a one-way function. To derive w from W= (α^w mod P') mod Q is at least to solve the discrete logarithm problem. Therefore, only the original signer can generate (w, r) to authorize someone as his proxy signer.

The integrity problem of the warrant w is discussed in the following. Given a (w, r), the proxy signer cannot replaces w with a new warrant w'. For the new warrant w', the proxy signer may obtains another W'= ($\alpha^{w'}$ mod P') mod Q. To generate the new signature (r, s') for W' is hard for the proxy signer since he does not have the secret key of the original signer. The proxy signer may try to find a warrant w' such that W= (α^w mod P') mod Q = ($\alpha^{w'}$ mod P') mod Q. So w'≡ w (mod P'-1). There are many solutions satisfying w'≡ w (mod P'-1) but only some limited solutions may be suitable for the warrant because the length of the warrant is limited. Moreover, the remaining solutions can be filtered out with the help of the set of formats of warrants since the warrant has to be written in a set of formats and contains data in special format. Therefore, the probability that w' is legal is insignificant.

The original signer cannot forge a proxy signature because he does not have the secret key of the proxy signer. Moreover, this scheme does not need secure channels in the proxy key generation phase. Even though the attackers intercept (r, s') in the proxy key generation phase, he does not have the secret key of the proxy signer to generate the proxy secret key s= s'+a+rx_B mod Q.

During the proxy key generation phase, the proxy signer cannot force the original signer sign some illegal message m' for him. If the proxy certificate (r', s') is the of the message m, then $h(m') \equiv (r'')^{x_A^{-1}}$ (mod P), where h is the one-way hash function used by the proxy signer for the message m'. However, the proxy signer does not know the secret key x_A, so he cannot construct $r'' = (h(m'))^{x_A}$ mod P.

The new proxy signature scheme has many advantages. The overall security of the new scheme is purely based on the discrete logarithm problem because this scheme does not need additional one-way hash functions. The new scheme need no secure channel while the most proposed proxy signature schemes need. The first proxy scheme without secure channels is Zhang's scheme [13]. In Zhang's scheme, the proxy signer could force the original signer to sign a message m during the proxy certification generation [11]. Fortunately, this problem is overcome in our scheme. With the help of the warrant, the new scheme provides powerful function to specify the proxy details. The new scheme also provides a fair proxy scheme. In the new scheme, only the original signer can authorized someone as his proxy signer while only the proxy signer can generate proxy signatures. The original signer cannot forge the proxy signatures while the proxy signer cannot transfer his proxy authorization to someone. Therefore, this new scheme supports the fair protection for the original signer and the proxy signers.

4. CONCLUSIONS

In a proxy signature scheme, an original singer can authorized someone as his proxy signer. In order to describe the details of the proxy authorization, a warrant is used to generate the proxy certificate. To integrate the warrant into the proxy certificate, the proposed proxy signature schemes suggest using the one-way hash functions. However, the one-way hash functions may weaken the overall security of the proxy signature schemes because the security of the most one-way hash functions are based on the complexity of a repeated simple function [1]. Without adopting one-way hash functions, the overall security of the proxy signature scheme will be only based on some cryptographic hard problems.

Here, a new proxy signature scheme without one-way hash function is proposed. Therefore, the overall security of this new scheme is only based on the discrete logarithm problem. This also makes the security analysis of the new scheme is clear. Another advantage is that the new scheme does not need secure channels. The new scheme is fair for the original and proxy signers. The original signer cannot forge proxy signatures while the proxy signer cannot generate any proxy signature without the authorization of the original signer. The new scheme also provides powerful specifying function to describe all of the proxy details by warrants. Moreover, the proxy signer cannot transfer the proxy authorization to another user without the agreement of the original signer.

REFERENCES

[1] Harn, L. (1997): " Digital Signature for Diffie-Hellman Public Keys without Using a One-way Function," Electronics Letters, Vol. 33, No. 2, 1997, pp. 125-126.

[2] Hwang, Sin-Jia and Shi, Chi-Hwai (1999): "The Specifiable Proxy Signature," NCS99, Vol 3, December 1999, pp. 190-197.

[3] Kim, S., Park, S., and Won, D. (1997):

"Proxy Signatures," ICICS ’97, Lecture Notes in Computer Science, Vol. 1334, Springer, Berlin, 1997, pp. 223-232.

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

[5] MAMBO, Masahiro, USUDA, Keisuke, and OKAMOTO, Eiji (1996a): "Proxy Signatures: Delegation of the Power to Sign Message," IEICE. Transaction Fundamentals, Vol. E 79-A, No. 9, Sept. 1996, pp.1338-1354.

[6] MAMBO, Masahiro, USUDA, Keisuke, and OKAMOTO, Eiji (1996b): "Proxy Signatures for Delegation Signing Operation," Proceedings of third ACM Conference on Computer and Communications Security, New Delhi, Mar. 1996, pp. 48-57.

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

[8] Sun, Hung-Min (1999): "An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," Computer Communications, Vol. 22, 1999, pp. 717-722.

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

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

[11] Sun, Hung-Min, Lee, N-Y and Hwang T. (1999): "Threshold Proxy Signatures," IEE Proc.-Computers and Digital Techniques, Vol. 146, No. 5, 1999, pp. 259-263.

[12] Varadharajan, V., Allen, P., and Black, S. (1991): "An Analysis of the Proxy Problem in Distributed Systems," Proceedings of 1991 IEEE Computer Society Symposium on Research m Security and Privacy, 1991, pp. 225-275.

[13] Zhang, K. (1997): "Threshold Proxy Signature Schemes," 1997 Information Security Workshop, Japan, Sep., 1997, pp. 191-197.