AN IMPROVEMENT OF STRONG PROXY SIGNATURE AND ITS APPLICATIONS

Min-Shiang Hwang
Department of Management Information Systems, National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.

Shiang-Feng Tzeng
Department of Computer Science and Information Engineering, National Central University
No. 300, Jung-da Rd., Jung-li City, 320 Taoyuan, Taiwan, R.O.C.

Shu-Fen Chiou
Department of Computer Science and Engineering, National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.

Keywords: Cryptography, digital signature, proxy signature, multi-proxy signature.

Abstract: In 2001, Lee et al. proposed a strong non-designated proxy signature for the use of multi-proxy signatures at the presence of plural delegations of multiple original signers. In this paper, we shall analyze their schemes and offer some suggestions as to how to improve the security of those schemes.

1 INTRODUCTION

Proxy signature schemes (Mambo et al., 1996a), (Mambo et al., 1996b) are what original signers can use to delegate their signing capability to so-called proxy signers. In these schemes, a proxy signature key is created by using the original signer's signature key. Then the proxy signer creates a signature to sign on behalf of the original signer. Several proxy signature schemes have been widely studied (Das et al., 2007), (Gu et al., 2005), (Guo and Liu, 2006), (Hwang et al., 2000), (Kim et al., 1997), (Petersen and Horster, 1997), (Sun, 1999), (Tzeng et al., 2002). In 2001, Lee et al. proposed a proxy signature scheme (Lee et al., 2001b). They have considered a number of possible attacks on their predecessors in their scheme.

Based on the types of weaknesses, Lee et al. (Lee et al., 2001b) have classified proxy signatures into strong and weak ones in terms of undeniability. A strong proxy signature can work both as an original signer's signature and as a proxy signer's signature, while a weak proxy signature can only act as an original signer's signature.

In addition, Lee et al. (Lee et al., 2001b) have also classified proxy signatures into designated and non-designated ones in terms of the designation of the proxy signer. They have shown that a strong proxy signature can be used without any proxy signer being designated, because the proxy signature has explicit authentic information about the proxy signer. Based on the above classifications, Lee et al. have proposed a Strong Non-designated Proxy Signature (SNPS) scheme and applied it to multi-proxy signature schemes in which multiple original signers delegate their signing capabilities to unspecified proxy signers.

In this article, we shall show various attacks on the above SNPS scheme and the SNPS-implemented multi-proxy signature scheme. Those schemes cannot satisfy the strong unforgeability requirement. Any third party or original signer is not designated as a proxy signer and thus is not allowed to create a valid proxy signature of the proxy signer. However, the original signer, or one of the original signers, can forge a valid proxy signature for the proxy signer which the proxy signer cannot repudiate.

In Sections 2 and 3, we shall review the SNPS scheme and observe how it can be applied to the construction of a multi-proxy signature scheme, and we shall also point out their weaknesses, respectively. In Section 4, our improved schemes and the security analysis of the improved schemes will be proposed and presented. Finally, the concluding remarks will

Hwang M., Tzeng S. and Chiou S. (2008).
AN IMPROVEMENT OF STRONG PROXY SIGNATURE AND ITS APPLICATIONS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 95-98
DOI: 10.5220/0001919800950098

be in the last section.

2 THE SNPS SCHEME AND ITS APPLICATION

In this section, we shall briefly review the Lee et al.'s SNPS scheme (LKK-SNPS for short) (Lee et al., 2001b) and its contribution to a multi-proxy signature scheme (Lee et al., 2001b).

2.1 Review of the SNPS Scheme

There are three phases in the LKK-SNPS scheme: proxy key issuing, proxy signer signing, and proxy signature verifying. Initially, the system parameters are defined as follows.

Let p be a large prime, q be a prime factor of $p-1$, g be a generator of order $q \in Z_p^*$, and $h(\cdot)$ be a one-way hash function. The warrant m_w records the identity of the original signer and the valid delegation time, etc. m_w does not include the identity of any proxy signer.

Each user U_i owns a private key $x_i \in Z_q^*$ and corresponding public key $y = g^{x_i} \bmod p$, which are certified by the certificate authority (CA). Let U_o be the original signer and U_p be the proxy signer.

- *Proxy Key Issuing:* U_o chooses a random number k and computes $r = g^k \bmod p$ and $\sigma = x_o h(m_w, r) + k \bmod q$. The tuple ($m_w$, r, σ) is U_o's signature on m_w. U_o sends (m_w, r, σ) to U_p. After receiving (m_w, r, σ), U_p verifies by checking whether the following equation holds:
$$g^{\sigma} = y_o^{h(m_w,r)} r \bmod p. \qquad (1)$$
If it holds, U_p computes her/his proxy private key σ_p as
$$\sigma_p = \sigma + x_p \bmod q. \qquad (2)$$
- *Proxy Signer Signing:* If a message m conforms to m_w, U_p can generate a proxy signature on m as $s = S(\sigma_p, m)$ using her/his proxy private key σ_p, where $S(\cdot)$ is a general signature generation algorithm. The tuple (m, s, m_w, r, y_o, y_p) is a valid proxy signature.
- *Proxy Signature Verifying:* The verifier computes the corresponding proxy public key:
$$y = y_o^{h(m_w,r)} r y_p \bmod p.$$
The verifier can verify (m, s, m_w, r, y_o, y_p) by checking if $m \in \{m_w\}$ and $V(y, m, \sigma) \overset{?}{=}$ true, where $V(\cdot)$ is a general signature verification algorithm. If those expressions hold, the proxy signature (m, s, m_w, r, y_o, y_p) for m is valid.

2.2 Review of the Multi-Proxy Signature Scheme

Let $G = \{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$ be the original group of n original signers. Now, they are trying to delegate their signing capabilities to some unspecified proxy signers. First, they can perform the same steps as the proxy key issuing phase in SNPS scheme. Each $U_{o_i} \in G$ sends (m_{w_i}, r_{o_i}, σ_i) to U_p.

After receiving (m_{w_i}, r_{o_i}, σ_i), U_p verifies it by Equation (1). If U_p wants to create a proxy signature on behalf of G under warrants $\{m_{w_1}, m_{w_2}, \cdots, m_{w_n}\}$, she/he has to generate her/his proxy private key σ_p as
$$\sigma_p = \sigma_1 + \cdots + \sigma_n + x_p \bmod q. \qquad (3)$$

If her/his message m conforms to $\{m_{w_1}, m_{w_2}, \cdots, m_{w_n}\}$, U_p can create a proxy signature on m as $s = S(\sigma_p, m)$. The tuple $(m, s, m_{w_1}, r_{o_1}, y_{o_1}, \cdots, m_{w_n}, r_{o_n}, y_{o_n}, y_p)$ is a valid proxy signature.

Then, any verifier can generate the proxy public key y as
$$y = y_{o_1}^{h(m_{w_1}, r_{o_1})} r_{o_1} \cdots y_{o_n}^{h(m_{w_n}, r_{o_n})} r_{o_n} y_p \bmod p.$$

Then, the verifier can check the validity of proxy signature by examining if $V(y, m, s) \overset{?}{=}$ true and $m \in \{m_{w_1}, m_{w_2}, \cdots, m_{w_n}\}$.

3 CRYPTANALYSIS

In this section, we shall analyze the security of the SNPS scheme and the SNPS-implemented multi-proxy signature scheme.

3.1 Cryptanalysis of the SNPS Scheme

In this subsection, we will show that the LKK-SNPS scheme is vulnerable to the public key substitution and direct forgery attacks. The original signer can generate a valid proxy signature key σ_p with respect to an arbitrary user. Let the arbitrary user be some proxy signer U_p.

In the public key substitution attack, U_o can make the public key substitution attack feasible. U_o selects a random number $k \in Z_q$, and computes $r = g^k \bmod p$. Then, she/he selects a random number $\alpha \in Z_q$ and updates her/his public key y_o by $y_o = g^{\alpha}(y_p^{-h(m_w,r)^{-1}}) \bmod p$. Thus, the valid proxy signature key is $\sigma_p = \alpha h(m_w, r) + k \bmod q$. The following expressions show why σ_p is valid.

$$\begin{array}{rcl} y & = & y_o^{h(m_w,r)} r y_p = (g^{\alpha}(y_p^{-h(m_w,r)^{-1}}))^{h(m_w,r)} r y_p, \\ & = & g^{\alpha h(m_w,r)} r = g^{\sigma_p} \bmod p. \end{array}$$

Finally, U_o can forge a valid proxy signature (m, s, m_w, r, y_o, y_p). In fact, U_p has never signed the message m, but she/he cannot deny it.

In the direct forgery attack, U_o randomly selects a number $k \in Z_q$ and computes $r = g^k y_p^{-1} \bmod p$. Then, she/he computes a valid proxy signature key $\sigma_p = x_o h(m_w, r) + k \bmod q$ because

$$\begin{aligned} y &= y_o^{h(m_w,r)} r y_p = y_o^{h(m_w,r)} (g^k y_p^{-1}) y_p, \\ &= y_o^{h(m_w,r)} g^k = g^{\sigma_p} \bmod p. \end{aligned}$$

Similarly, U_o can forge a valid proxy signature, and U_p cannot deny to signing the message m.

3.2 Cryptanalysis of the Multi-Proxy Signature Scheme

In this subsection, we will show that the Lee-Kim-Kim multi-proxy signature is vulnerable to the collusion attack, the public key substitution attack, and the direct forgery attack. Cooperation of all the original signers or one malicious original signer can forge valid multi-proxy signatures.

Without loss of generality, suppose $\{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$ want to forge a multi-proxy signature on m for an arbitrarily chosen proxy signer U_p by collusion attack. U_{o_1} first selects k_{o_1} and computes $r_{o_1} = g^{k_{o1}} y_p^{-1} \bmod p$ and $\sigma_1 = x_{o_1} h(m_{w_1}, r_{o_1}) + k_{o_1} \bmod q$. Thus, the valid multi-proxy signature key is $\sigma_p = \sigma_1 + \cdots + \sigma_n \bmod q$. The following expressions show why σ_p is valid.

$$\begin{aligned} y &= y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p, \\ &= y_{o_1}^{h(m_{w_1},r_{o_1})} (g^{k_{o1}} y_p^{-1}) \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p, \\ &= y_{o_1}^{h(m_{w_1},r_{o_1})} g^{k_{o1}} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n}, \\ &= g^{\sum_{i=1}^{n} \sigma_i} \bmod p. \end{aligned}$$

Therefore, all the original signers can work together and use σ_p to generate a forged multi-proxy signature on an arbitrary message m for an arbitrary proxy signer U_p.

In the public key substitution attack, any original signer can forge valid multi-proxy signatures by updating her/his own public key. Suppose that U_{o_1} wants to forge a multi-proxy signature on m for $\{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$. This attack is similar in Section 3.1. U_{o_1} first selects random numbers m_{w_i}, k_{o_i} for $i = 1, 2, \cdots, n$, and α, and then she/he computes $r_{o_i} = g^{k_{o_i}} \bmod p$ and $y_{o_1} = g^{\alpha} (y_{o_2}^{h(m_{w_2},r_{o_2})} r_{o_2} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p)^{-h(m_{w_1},r_{o_1})^{-1}} \bmod p$. Then, U_{o_1} makes a request to CA for updating her/his public key by y_{o_1}. Thus, the valid proxy signature key is $\sigma_p = \alpha h(m_{w_1}, r_{o_1}) + k_{o_1} \bmod q$, and its corresponding proxy public key is $y = g^{\sigma_p} \bmod p$. This is because

$$\begin{aligned} y &= y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p, \\ &= (g^{\alpha} (y_{o_2}^{h(m_{w_2},r_{o_2})} r_{o_2} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p)^{-h(m_{w_1},r_{o_1})^{-1}})^{h(m_{w_1},r_{o_1})} \\ & \quad r_{o_1} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p = g^{\alpha h(m_{w_1},r_{o_1})} r_{o_1} = g^{\sigma_p} \bmod p. \end{aligned}$$

Therefore, U_{o_1} can use σ_p to generate a forged multi-proxy signature on an arbitrary message m for $\{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$.

In the directing forgery attack, we assume U_{o_1} wants to forge a multi-proxy signature on m for $\{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$. The forgery attack is also similar in Section 3.1. U_{o_1} first selects random numbers m_{w_i} and k_{o_i} for $i = 1, 2, \cdots, n$, and then she/he computes

$$r_{o_1} = g^{k_{o1}} (y_{o_2}^{h(m_{w_2},r_{o_2})} r_{o_2} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p)^{-1} \bmod p.$$

Then the valid proxy signature key is $\sigma_p = x_{o_1} h(m_{w_1}, r_{o_1}) + k_{o_1} \bmod q$, and its corresponding proxy public key is $y = g^{\sigma_p} \bmod p$. This is because

$$\begin{aligned} y &= y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p, \\ &= y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} (y_{o_2}^{h(m_{w_2},r_{o_2})} r_{o_2} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_{o_n} y_p)^{-1} \\ & \quad \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} r_n y_p = y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} = y^{\sigma_p} \bmod p. \end{aligned}$$

Therefore, U_{o_1} can use σ_p to generate a forged multi-proxy signature on an arbitrary message m for $\{U_{o_1}, U_{o_2}, \cdots, U_{o_n}\}$.

4 OUR IMPROVEMENT

In this section, we modify the SNPS scheme and its multi-proxy signature scheme to remedy the weaknesses described in Section 3.

4.1 The Improved Schemes

In the SNPS scheme, the proxy signature can be forged by the original signer. We modify the scheme as follows. In the proxy signer signing phase, we turn Equation (2) into

$$\sigma_p = \sigma + x_p h(m_w, r, y_o) \bmod q.$$

Therefore, the proxy public key y becomes

$$y = y_o^{h(m_w,r)} r y_p^{h(m_w,r,y_o)} \bmod p.$$

To remedy the weaknesses of the multi-proxy signature scheme, we shall treat it similarly. The new σ_p in Equation (3) is

$$\sigma_p = \sum_{i=1}^{n} \sigma_i + x_p h(m_{w_1}, r_{o_1}, y_{o_1}, \cdots, m_{w_n}, r_{o_n}, y_{o_n}) \bmod q.$$

Therefore, the proxy public key y becomes

$$\begin{aligned} y &= y_{o_1}^{h(m_{w_1},r_{o_1})} r_{o_1} \cdots y_{o_n}^{h(m_{w_n},r_{o_n})} \\ &\quad r_{o_n} y_p^{h(m_{w_1},r_{o_1},y_{o_1},\cdots,m_{w_n},r_{o_n},y_{o_n})} \bmod p. \end{aligned}$$

4.2 Security Analysis

The improved schemes can withstand all the above attacks in Section 3. In the SNPS scheme, suppose the signer U_o is a malicious original signer. U_o selects a random integer α and makes her/his public key y'_o satisfy the following equation

$$y'_o = g^{\alpha}(y_p^{-h(m_w,r,y_o)^{-1}}) \bmod p.$$

If U_o fixes the integer y'_o, she/he will have to solve the discrete logarithm problem to find the value of α; on the other hand, if U_o first determines the integer α, then she/he has to obtain the value of y'_o by solving the difficult problem. Therefore, the public key substitution attack is not likely to work.

As for the directing forgery attack, the security analysis is the same as that of the public key substitution attack on the improved schemes. The proxy signature cannot be forged by direct forgery attack. Therefore, those attacks on the improved SNPS scheme and its application to multi proxy signatures are impossible since it is difficult to obtain the proxy signature.

5 CONCLUSIONS

In this paper, we have shown that strong non-designated proxy signature schemes and their applications to multi-proxy signature schemes are vulnerable to some attacks. The malicious original signer can forge valid strong non-designated proxy signatures and multi-proxy signatures. Furthermore, the proxy signer cannot repudiate the forged proxy signatures. Therefore, we have also presented our improved scheme to defeat those attacks.

Lee et al. have also presented several mobile applications of strong proxy signatures. In (Lee et al., 2001a), Lee et al. have shown that mobile agents can be constructed by using strong non-designated proxy signatures. However, the same attacks on strong non-designated proxy signatures can be generalized to work on Lee-Kim-Kim “secure” mobile agents. Again, our improved scheme can be used here to defeat these attacks.

ACKNOWLEDGEMENTS

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 96-2219-E-001-001, and NSC 96-2219-E-009-013.

REFERENCES

Das, M. L., Saxena, A., and Phatak, D. B. (2007). Proxy signature scheme with effective revocation using bilinear pairings. *International Journal of Network Security*, 4(3):312–317.

Gu, L. Z., Zhang, S., and Yang, Y. X. (2005). An improved proxy multi-signature scheme. *The Journal of China Universities of Posts and Telecommunications*, 12(1):10–14.

Guo, L. and Liu, Y. (2006). Security analysis and improvement of hsu et al. threshold proxy signature scheme. *International Journal of Network Security*, 2(1):69–72.

Hwang, M. S., Lin, I. C., and Lu, E. J. L. (2000). A secure nonrepudiable threshold proxy signature scheme with known signers. *International Journal of Informatica*, 11(2):1–8.

Kim, S., Park, S., and Won, D. (1997). Proxy signatures, revisited. *Proc. of ICICS'97, LNCS 1334*, pages 223–232.

Lee, B., Kim, H., and Kim, K. (2001a). Secure mobile agent using strong non-designated proxy signature. In *Lecture Notes in Computer Science 2119, ACISP 01*, pages 474–486, Sydney, Australia.

Lee, B., Kim, H., and Kim, K. (2001b). Strong proxy signature and its applications. In *The 2001 Symposium on Cryptography and Information Security*, pages 603–608, Oiso, Japan.

Mambo, M., Usuda, K., and Okamoto, E. (1996a). Proxy signatures: Delegation of the power to sign message. *IEICE Trans. Fundamentals*, E79-A(9):1338–1353.

Mambo, M., Usuda, K., and Okamoto, E. (1996b). Proxy signatures for delegating signing operation. *Proc. Third ACM Conf. on Computer and Communications Security*, pages 48–57.

Petersen, H. and Horster, P. (1997). Self-certified keys - concepts and applications. In *Communications and Multimedia Security'97*, pages 102–116, Chapman & Hall.

Sun, H. M. (1999). An efficient nonrepudiable threshold proxy signature scheme with known signers. *Computer Communications*, 22(8):717–722.

Tzeng, S.-F., Yang, C.-Y., and Hwang, M.-S. (2002). A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification. *Proceeding of* 12^{th} *National Conference on Information Security, R.O.C.*, pages 285–292.