Cryptanalysis of a threshold proxy signature with known signers

Fuw-Yi Yang, Jinn-Ke Jan[*], and Woei-Jiunn Jeng[*]

Department of Applied Mathematics, National Chung Hsing University

Taichung Taiwan 402, R.O.C, E-mail: yangfy@ms7.hinet.net

[*]Department of Computer Science, National Chung Hsing University

Taichung Taiwan 402, R.O.C, E-mail: jkjan@cs.nchu.edu.tw

Abstract

A scheme of threshold proxy signature with known signers was proposed by Hwang *et al*. In their scheme, the receiver can identify the proxy signers that actually generated a proxy signature. Tzeng *et al.* demonstrated that this signature scheme is insecure and proposed an improvement to mend the information leakage. This paper shows that the improved scheme is still insecure under the original signer's forgery attack.

Keywords

Digital signature, non-repudiation, proxy signature, threshold proxy signature.

Introduction

Mambo *et al.* proposed proxy signatures in 1996 [2, 3]. By the scheme of proxy signature, an original signer can delegate her/his signing capacity to a proxy signer. Therefore the proxy signer can issue proxy signatures, which are signatures signed by the proxy signer on behalf of the original signer. Then, the proxy signatures are verified using the public parameters of the proxy signer and original signer.

A scheme of *(t, n)* threshold proxy signature is an integration of *(t, n)* threshold scheme with proxy signature. Among a set of *n* predetermined proxy members (signers), *t* or more than *t* proxy members can construct proxy signatures on behalf of the original signer. On the other hand, less than *t* proxy members cannot construct any valid proxy signature.

Hwang *et al.* proposed a *(t, n)* threshold proxy signature scheme (henceforth called HLL-scheme) with known signers [1]. In HLL-scheme, proxy signatures are verified using the public parameters of the *n* proxy members, *t* proxy signers (actual signers), and original signer. Thus the verifier can identify the proxy signers that actually constructed the signatures. However, Tzeng *et al.* have successfully mounted an insider attack on HLL-scheme [4]. In their attack, the original signer can forge proxy signatures without the assistance of the proxy signers. They also proposed an improvement (THY-scheme) to remedy this weakness in security.

This paper demonstrates a new insider attack to show that both the HLL-scheme and the THY-scheme are insecure. Under the new insider attack, the original signer can forge proxy signatures on any messages. Furthermore, the original signer can change the content of a delegation warrant, which is a portion of the proxy signature. The delegation warrant contains information of delegated signing information, *e.g.* the *n* proxy members, the original signer, the threshold value *t*, and the valid delegation time. The result of modifiable content in delegation warrant would cause damage to the proxy signers, since the original signer can shorten or lengthen the expiration date of a delegation warrant.

Review of HLL-scheme

Let p be a large prime number such that $(p - 1)$ has a large prime factor q. The element g in the group Z_p^* has order q. $e \in_R G$ represents that the element e is randomly chosen from the group G. $|b|$ denotes the bit length of the string b. $|B|$ represents the number of elements (members) in the set B. $h(.) : \{0, 1\}^* \rightarrow \{0, 1\}^l$ is a collision-free hash function, where l is a security parameter, *i.e.* $l = 160$ or $l = |q|$. *ASID* denotes the identities of the actual signers. The symbol m_w represents a delegation warrant. The content of a delegation warrant includes the identity of the original signer and proxy signers, threshold parameters t and n, and expiration date.

The original signer p_o has a private key $x_o \in_R Z_q^*$ and the corresponding public key $y_o = g^{x_o} \mod p$. $G = \{p_1, p_2, \ldots, p_n\}$ is a group of n proxy members (proxy signers). Like the original signer, each proxy member $p_i \in G$ has a private key $x_i \in_R Z_q^*$ and the public key $y_i = g^{x_i} \mod p$. Let $D \subseteq G$ be a set of actual proxy signers. If $|D|$ is equal to or larger than the threshold value t, *i.e.* $t \le |D|$, then the proxy signers in the set D can cooperatively construct proxy signatures. The following steps describe the details. Please note that the members in G are called proxy members and the members in D are called proxy signers.

Step 1. Key generation of proxy group

The purpose of this step is to construct a secret polynomial of degree $t - 1$. The polynomial is used to distribute secret share among the members of G. Thus a set of t or more members can recover the secret.

Each proxy member $p_i \in G$ generates a polynomial $f_i(x) = (x_i + \sum_{k=0}^{t-1} a_{i,k} x^k) \mod q$ and computes the quantities $A_{i,k} = g^{a_{i,k}} \mod p$, where $a_{i,k} \in_R Z_q$ and $k = 0, 1, \ldots, (t-1)$. For every $p_j \in G$, the proxy member p_i computes the quantity $f_i(j)$. Then p_i sends the quantities $f_i(j), A_{i,0}, A_{i,1}, \ldots,$ and $A_{i,t-1}$ to p_j. Thus every proxy member p_i obtains the secret value s_i and public values $A_0, A_1, \ldots, A_{t-1}$ and Y_G as follows.

$$s_i = f(i) = f_1(i) + \ldots + f_n(i)$$

$$= (\sum_{k=1}^{n} x_k + \sum_{k=0}^{t-1} a_k i^k) \mod q, \qquad (1)$$

$$Y_G = y_1 y_2 \ldots y_n \mod p,$$

$A_0 = g^{a_0} = A_{1,0}\, A_{2,0} \ldots A_{n,0}\ mod\ p,$

$A_1 = g^{a_1} = A_{1,1}\, A_{2,1} \ldots A_{n,1}\ mod\ p, \ldots,$ and

$A_{t-1} = g^{a_{t-1}} = A_{1,t-1}\, A_{2,t-1} \ldots A_{n,t-1}\ mod\ p$, where $a_k = \sum_{j=1}^{n} a_{j,k}$ and $k = 0, 1, \ldots, (t-1)$.

The proxy members check and publish $A_0, A_1, \ldots, A_{t-1}$ and Y_G.

Step 2. Original signer generates proxy key

The original signer constructs a delegation warrant m_w and proxy key $\sigma = (x_o\, h(m_w, K) + k)\ mod\ q$, where $K = g^k\ mod\ p$ and $k \in_R Z_q$. The proxy key is actually a signature of the original signer on the delegation warrant m_w.

Step 3. Distribute the proxy key among the *n* proxy members

The original signer shares out the proxy key σ among the n proxy members such that any t or more than t proxy members can reconstruct the proxy key. The original signer performs the following sub-steps to distribute the proxy key shares among the n proxy members.

3.1 Generate a polynomial $f'(x) = (\sigma + \sum_{k=1}^{t-1} b_k x^k)\ mod\ q$, where $b_k \in_R Z_q$.

3.2 Publish $B_1, \ldots, B_{t-1}, m_w$, and K, where $B_k = g^{b_k}\ mod\ p$ and $k = 1, 2, \ldots, (t-1)$.

3.3 For each proxy member $p_i \in G$, the original signer computes the quantity $\sigma_i = f'(i)$ and sends σ_i to the proxy member p_i in a secure way.

3.4 Upon receiving σ_i, p_i verifies it by checking $g^{\sigma_i} = y_o^{h(m_w, K)} K \prod_{k=1}^{t-1} (B_k)^{i^k}\ mod\ p$.

If the check is valid, the proxy member p_i constructs the proxy key share (proxy signing key) σ'_i as follows:

$\sigma'_i = (\sigma_i + s_i\, h(m_w, K))\ mod\ q.$

Step 4. Proxy signature generation

Let $D = \{p_1, p_2, \ldots, p_t\}$ be a subset of G. Assume that the proxy signers in the set D have agreed to sign on the message m on behalf of the original signer. The following sub-steps describe the details of issuing proxy signatures.

4.1 Like step 1, each proxy signer $p_i \in D$ generates a polynomial $f''_i(x) = (x_i + \sum_{k=0}^{t-1} c_{i,k} x^k)\ mod\ q$ and computes the quantities $C_{i,k} = g^{c_{i,k}}\ mod\ p$, where $c_{i,k} \in_R Z_q$ and $k = 0, 1,\ldots, (t-1)$. For every $p_j \in D$, the proxy signer p_i computes the quantity $f''_i(j)$ and sends $f''_i(j), C_{i,0,}, C_{i,1},\ldots, C_{i,t-1}$ to p_j. Thus the proxy signer p_i obtains the secret value s'_i and public values $C_0, C_1,\ldots, C_{t-1}$ and Y as follows.

$s'_i = f''(i) = f''_1(i) + \ldots + f''_t(i)$

$$= (\sum_{k=1}^{t} x_k + \sum_{k=0}^{t-1} c_k i^k)\ mod\ q, \tag{2}$$

$Y = C_0 = g^{c_0} = C_{1,0}\, C_{2,0} \ldots C_{t,0}\ mod\ p,$

$C_1 = g^{c_1} = C_{1,1}\, C_{2,1} \ldots C_{t,1}\ mod\ p, \ldots,$ and

$C_{t-1} = g^{c_{t-1}} = C_{1,t-1}\, C_{2,t-1} \ldots C_{t,t-1}\ mod\ p,$ where $c_k = \sum_{j=1}^{t} c_{j,k}$ and $k = 0, 1,\ldots, (t-1).$

These proxy signers check and publish $C_1,\ldots, C_{t-1}$ and Y.

4.2 Each $p_i \in D$ computes the quantity $\gamma_i = (s'_i\, Y + \sigma'_i\, h(ASID, m))\ mod\ q$ and sends γ_i to each $p_j \in D$ except for p_i himself.

4.3 On receiving γ_j, p_i verifies γ_j by checking $g^{\gamma_j} = [Y(\prod_{k=1}^{t-1}(C_k)^{j^k})(\prod_{k=1}^{t} y_k)]^Y [y_o^{h(m_w,K)} K(\prod_{k=1}^{t-1}(B_k)^{j^k})(Y_G A_0 \prod_{k=1}^{t-1}(A_k)^{j^k})^{h(m_w,K)})]^{h(ASID,\ m)}\ mod\ p$. After verifying all γ_j for $j = 1, 2,\ldots, t$, all the proxy signers in the set D cooperatively reconstruct the values $f(0), f'(0),$ and $f''(0)$ using $f(i), f'(i)$ and $f''(i)$ respectively. Then, these t proxy signers compute the quantity

$$T = [f''(0)Y + (f'(0) + f(0))\ h(ASID,\ m)]\ mod\ q \tag{3}$$

and issue the proxy signature $(m, T, K, Y, A_0, m_w, ASID)$.

Step 5. Proxy signature verification

The receiver verifies the proxy signature $(m, T, K, Y, A_0, m_w, ASID)$ using equation

$$g^T = [\, y_o^{h(m_w,K)} KA_0 (\prod_{k=1}^{n} y_k)\,]^{h(ASID,\, m)}\ [Y(\prod_{k=1}^{t} y_k)\,]^Y \mod p. \qquad (4)$$

Review of THY-scheme (Improvement of HLL-scheme)

Their improvement, the THY-scheme, was obtained by replacing equation (1) in step 1 with equation (5) and replacing equation (2) in step 4.1 with equation (6) below.

$$s_i = f(i) = \sum_{k=1}^{n} x_k y_k + a_0 A_0 + \sum_{k=1}^{t-1} a_k i^k \mod q \qquad (5)$$

$$s'_i = f''(i) = \sum_{k=1}^{t} x_k y_k + c_0 C_0 + \sum_{k=0}^{t-1} c_k i^k \mod q \qquad (6)$$

Thus, the improved proxy signature is verified as follows.

$$g^T = (y_o^{h(m_w,K)} KA_0^{A_0} \prod_{i=1}^{n} y_i^{y_i})^{h(ASID,m)} (Y^Y \prod_{i=1}^{t} y_i^{y_i})^Y \mod p$$

Cryptanalysis of THY-scheme and HLL-scheme

We first show an insider attack on the THY-scheme. Then demonstrate that the same attack is workable on the HLL-scheme. The original signer chooses a random number $a \in_R Z_q$, message m' and delegation warrant m'_w. After choosing the message and delegation warrant in demand, the original signer computes the quantities

$K' = g^{a / h(ASID,\, m')} (A_0^{A_0} \prod_{i=1}^{n} y_i^{y_i})^{-1} (Y^Y \prod_{i=1}^{t} y_i^{y_i})^{-Y / h(ASID,m')} \mod p$ and

$T' = (a + x_o\, h(m'_w, K')\, h(ASID, m')) \mod q.$

Then, the original signer has forged the proxy signature $(m', T', K', Y, A_0, m'_w, ASID)$. The counterfeited proxy signature is verified as follows.

$$g^{T'} = g^{a + x_o h(m'_w,K') h(ASID,m')} = g^a (y_o^{h(m'_w,K')})^{h(ASID,m')}$$

$$= (K' \ (A_0^{A_0} \prod_{i=1}^{n} y_i^{y_i}))^{h(ASID,\, m')} (Y^Y \prod_{i=1}^{t} y_i^{y_i})^Y \ (y_o^{h(m'_w,K')})^{h(ASID,m')}$$

$$= (\, y_o^{h(m'_w,K')} K' A_0^{A_0} \prod_{i=1}^{n} y_i^{y_i} \ )^{h(ASID,\, m')} (Y^Y \prod_{i=1}^{t} y_i^{y_i})^Y \mod p$$

Subsequently, we demonstrate the same attack on the HLL-scheme. After choosing the random number a, message m', and delegation warrant m'_w, the original signer computes the quantities

$K' = g^{a / h(ASID,\, m')} (A_0 \prod_{i=1}^{n} y_i)^{-1} (Y \prod_{i=1}^{t} y_i)^{-Y / h(ASID, m')}\ mod\ p$ and

$T' = (a + x_o\, h(m'_w, K')\, h(ASID, m'))\ mod\ q$.

Then, the adversary has forged the proxy signature $(m', T', K', Y, A_0, m'_w, ASID)$. The counterfeited proxy signature is verified as follows.

$$g^{T'} = g^{a + x_o h(m'_w, K') h(ASID, m')} = g^a (y_o^{h(m'_w, K')})^{h(ASID, m')}$$

$$= (K' A_0 \prod_{i=1}^{n} y_i)^{h(ASID,\, m')} (Y \prod_{i=1}^{t} y_i)^{Y} (y_o^{h(m'_w, K')})^{h(ASID, m')}$$

$$= (y_o^{h(m'_w, K')} K' A_0 \prod_{i=1}^{n} y_i)^{h(ASID,\, m')} (Y \prod_{i=1}^{t} y_i)^{Y}\ mod\ p$$

Conclusion

We have shown that both the HLL-scheme and THY-scheme are vulnerable to the insider attack.

References

1. M. S. Hwang, I. C. Lin, and Eric J. L. Lu, “A secure nonrepudiable threshold proxy signature scheme with known signers,” *International Journal of Informatica*, Vol. 11(2), pp. 1-8, 2000.

2. M. Mambo, K. Usuda, and E. Okamoto, “Proxy signatures: delegation of the power to sign message,” *IEICE Transactions on Fundamentals*, Vol. E79-A, pp. 1338-53, 1996a.

3. M. Mambo, K. Usuda, and E. Okamoto, “Proxy signatures for delegating signing operation,” *Proceedings of the Third ACM Conference on Computer and Communications Security,* pp. 48-57, 1996b.

4. S. F. Tzeng, M. S. Hwang, C. Y. Yang, “An improvement of nonrepudiable threshold proxy signature scheme with known signers,” *Computers & Security,* Vol.

23, pp. 174-178, 2004.