Generalization of Proxy Signature Based on Discrete Logarithms*

Li-Hua Li[‡] Shiang-Feng Tzeng[‡] Min-Shiang Hwang[†]

Graduate Institute of Networking and Communication Engineering[†]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-23742337

Department of Information Management[‡]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.

October 23, 2002

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC91-2213-E-324-003.

[†]Responsible for correspondence: Prof. Min-Shiang hwang

Generalization of Proxy Signature Based on Discrete Logarithms

Abstract

In the past few years, many excellent studies on proxy signature schemes have been published. Yet, traditional proxy signature schemes are mainly aimed at dealing with one or two separate proxy conditions each. In this article, the authors shall present a generalized version of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on the discrete logarithms that can be applied to every proxy situation. The $(t_1/n_1 - t_2/n_2)$ proxy signature scheme allows the original group of original signers to delegate their signing capability to a designated proxy group. The proxy group of proxy signers can cooperatively generate the proxy signature on behalf of the original group. Any verifier can verify the proxy signature on the message with the knowledge of the identities of the actual original signers and the actual proxy signers. Furthermore, some possible attacks have been considered, and our security analysis shows that none of them can successfully break the proposed scheme.

Keywords: Digital signature, Proxy signature, Threshold proxy signature, Proxy multi-signature scheme.

1 Introduction

Digital signatures [2, 14] have been more and more commonly used in place of hand-written signatures in the digital world. However, most digital signature schemes are not applicable to all particular situations. For example, suppose a manager in a company needs to go on a business trip. She/he has to find a proxy person to take care of her/his work at the office. Common digital signature schemes [6, 7, 8] do not meet this manager's need. To cover this situation, the proxy function has been added to digital signature schemes, and this new type of digital signature is called the proxy signature. The proxy signature [13] allows the original signer to delegate her/his signing capability to a designated proxy signer so that the designated proxy signer can generate a signature on behalf of the original signer.

So far, quite a few proxy signature schemes have been published and widely discussed [3, 4, 9, 10, 11, 16, 17, 18]. However, in those proxy signature schemes, the original signer can delegate only one proxy signer instead of a proxy group. As a result, to take the proxy group case in to consideration, some threshold proxy signature schemes ($(1-t_2/n_2)$ proxy signature schemes) and proxy multi-signature schemes ((n_1-1) proxy signature schemes) have also been proposed.

Recently, $(1-t_2/n_2)$ proxy signature schemes have drawn much attention [3, 4, 9, 10, 11, 18]. In such a design, the original signer is able to delegate her/his signing capability to n_2 proxy signers of the proxy group. Any t_2 or more proxy signers in the proxy group can cooperatively generate the proxy signature.

On the other hand, (n_1-1) proxy signature scheme can be found in [17]. In the scheme, n_1 original signers in the original group can cooperatively delegate their signing capability to a proxy singer. The proxy signer is then allowed to create a proxy signature on behalf of the original group. However, the scheme has a common disadvantage: the size of the proxy signature is proportional to the number of the original signers.

In [15], Sun presents the concept of $(1-t_2/n_2)$ proxy signature with known signers. Any t_2 or more signers from the group who have actually signed the messages are unknown and cannot be identified. This is very convenient for auditing purposes. However, to place the responsibility on the actual signers and trace the dishonest signers if any, it would be necessary to identify who the actual signers are. Afterwards, the studies in [3, 4] are aimed at discussing the security.

The above proxy signature schemes are mainly aimed at dealing with one or two separate proxy situations each. In this article, we shall propose a generalized $(t_1/n_1 - t_2/n_2)$ proxy signature scheme with known signers and focus on the proxy signature authorized by using partial delegation with warrant. The proposed scheme is suitable for all the proxy situations mentioned above. In reality, the $(n_1 - 1)$ proxy signature scheme is a special case of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme where the value t_1 equals n_1 and the proxy group has only one single proxy signer. On the other hand, the $(1 - t_2/n_2)$ proxy signature scheme is also a special case of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme where the original group has only one single original signer. As for the case where the original group includes only one original signer and the proxy group is made of only one proxy singer, it is also a special case of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme.

This article is organized as follows. In Section 2, a new secure $(t_1/n_1 - t_2/n_2)$ proxy signature scheme will be presented. In Section 3, we shall prove that the proposed scheme can work correctly. In Section 4, the security of the proposed scheme will be analyzed. After that, we shall discuss some special cases of the $(t_1/n_1 - t_2/n_2)$ proxy signature in Section 5. Finally, some concluding remarks will be in the last section.

2 The Proposed Scheme

In this section, we shall present a new $(t_1/n_1 - t_2/n_2)$ proxy signature scheme. The proposed scheme involves the following participants: the original group (G_O), the proxy group (G_P), the verifier and two designated clerks (C_O and C_P). The services provided by C_O for G_O are to collect the individual signatures generated by the original signers so as to construct a proxy share. In addition, C_O can verify the validity of these individual signatures from the actual original signers. Similarly, the services provided by C_P for G_P are to collect the individual proxy signatures generated by the proxy signers so as to construct the final proxy signature. Besides, C_P can verify the validity of these individual proxy signatures signed by the actual proxy signers.

According to the proxy warrant of the proposed scheme, any t_1 or more out of n_1 original signers $(1 \leq t_1 \leq n_1)$ can represent G_O to delegate the signing capability. Similarly, any t_2 or more out of n_2 proxy signers $(1 \leq t_2 \leq n_2)$ can represent G_P to sign a message on behalf of G_O. Any verifier can verify the proxy signature.

The system parameters of the proposed scheme are defined in Table 1.

Table 1: The system parameters of the proposed scheme

Parameter	Denotation
p	a large prime
q	a large prime factor of $p-1$
g	a generator in $GF(p)$ of order q
$h(\cdot)$	a one-way hash function
M_w	a warrant which records the identities of the original signers in G_O and the proxy signers in G_P, the parameters $(t_1,\, n_1)$ and $(t_2,\, n_2)$, and the valid delegation time, etc.
$AOSID$	the identities of the actual original signers
$APSID$	the identities of the actual proxy signers

Each user U_i has a randomly selected private key $x_i \in Z_q^*$ and a public key y_i certified by the certificate authority (CA).

$$y_i = g^{x_i} \bmod p. \tag{1}$$

Let $G_O = \{U_{O_1}, U_{O_2}, \cdots, U_{O_{n_1}}\}$ and $G_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{n_2}}\}$ be groups of n_1 original signers and n_2 proxy signers, respectively.

The proposed scheme can be divided into three phases: the proxy share generation phase, the proxy signature generation phase, and the proxy signature verification phase. In the proxy share generation phase, the original signers cooperatively generate the proxy share and send it to G_P. In the proxy signature generation phase, the proxy signers cooperatively generate a valid proxy signature for a message on behalf of G_O. In the proxy signature verification phase, the verifier checks the validity of the proxy signature and can identify not only the actual original signers but also the actual proxy signers.

2.1 Proxy Share Generation Phase

Suppose any t_1 or more original signers want to delegate their signing capability to G_P so that G_P can sign messages on behalf of G_O. Let $D_O = \{U_{O_1}, U_{O_2}, \cdots, U_{O_{t'_1}}\}$ be the actual original signers, where $t_1 \le t'_1 \le n_1$. D_O as a group executes the following steps to delegate the signing capability to G_P.

1. Select a random integer $t_{O_i} \in Z_q^*$ and broadcast k_{O_i} to the other $t'_1 - 1$ original signers in D_O and the designated clerk C_O.

$$k_{O_i} = g^{t_{O_i}} \bmod p. \tag{2}$$

2. After receiving k_{O_j} $(j = 1, 2, \cdots, t'_1; j \ne i)$, each $U_{O_i} \in D_O$ calculates K and σ_{O_i} as follows:

$$\begin{aligned} K &= \prod_{i=1}^{t'_1} k_{O_i} \bmod p, & (3) \\ \sigma_{O_i} &= t_{O_i} K + x_{O_i} y_{O_i} h(M_w, K, AOSID) \bmod q. & (4) \end{aligned}$$

3. Send σ_{O_i} to C_O via a public channel.

4. For each received σ_{O_i} $(i = 1, 2, \cdots, t'_1)$, C_O checks whether the following equation holds:

$$g^{\sigma_{O_i}} = k_{O_i}^K y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod p. \tag{5}$$

If it does, then C_O calculates σ_O as their proxy share:

$$\sigma_O = \sum_{i=1}^{t'_1} \sigma_{O_i} \bmod q. \tag{6}$$

5. Broadcast $(M_w, K, \sigma_O, AOSID)$ to G_P.

After receiving $(M_w, K, \sigma_O, AOSID)$, each $U_{P_j} \in G_P$ checks whether the following equation holds:

$$g^{\sigma_O} = K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod p. \tag{7}$$

If it does, each $U_{P_j} \in G_P$ confirms the validity of $(M_w, K, \sigma_O, AOSID)$. Then, each U_{P_j} uses σ_O as her/his proxy share.

2.2 Proxy Signature Generation Phase

Without loss of generality, the proposed scheme allows any t_2 or more proxy signers to represent G_P to sign a message M cooperatively on behalf of G_O. Let $D_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{t'_2}}\}$ be the actual proxy signers for $t_2 \le t'_2 \le n_2$. D_P as a group executes the following steps to generate a proxy signature.

1. Select a random integer $t_{P_j} \in Z_q^*$ and calculate r_{P_j} as follows:

$$r_{P_j} = g^{t_{P_j}} \bmod p.$$

   Next, send r_{P_j} to the other $t'_2 - 1$ proxy signers in D_P and the designated clerk C_P.

2. After receiving r_{P_k} $(k = 1, 2, \cdots, t'_2; k \ne j)$, each $U_{P_j} \in D_P$ calculates R and s_{P_j} as follows:

$$\begin{aligned} R &= \prod_{j=1}^{t'_2} r_{P_j} \bmod p, &(8)\\ s_{P_j} &= t_{P_j} R + (\sigma_O t_2'^{-1} + x_{P_j} y_{P_j})^{h(M,R,APSID)} \bmod q. &(9) \end{aligned}$$

   Here, s_{P_j} is an individual proxy signature which is sent to C_P.

3. For each received s_{P_j} $(j = 1, 2, \cdots, t'_2)$, C_P checks whether the following equation holds:

$$g^{s_{P_j}} = r_{P_j}^R ((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)})^{t_2'^{-1}} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p. \tag{10}$$

If it does, (r_{P_j}, s_{P_j}) is a valid individual proxy signature of M. If all the individual proxy signatures of M are valid, the designated clerk calculates

$$S = \sum_{j=1}^{t'_2} s_{P_j} \bmod q. \tag{11}$$

Then, the proxy signature of M is $(M_w, K, AOSID, M, R, S, APSID)$.

2.3 Proxy Signature Verification Phase

After receiving the proxy signature $(M_w, K, AOSID, M, R, S, APSID)$ for M, any verifier can verify the validity of the proxy signature by following the steps below:

1. According to M_w, $AOSID$ and $APSID$, the verifier obtains the public keys of the original signers and proxy signers from CA and knows whom the actual original signers and actual proxy signers are.

2. The verifier checks the validity of the proxy signature on the message M through the following equation:

$$g^S = R^R (K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i}}{}^{h(M_w,K,AOSID)} \prod_{j=1}^{t'_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p. \tag{12}$$

If the equation holds, the message M is authenticated, and the proxy signature $(M_w,\ K,\ AOSID,\ M,\ R,\ S,\ APSID)$ is valid.

3 Correctness of The Proposed Scheme

In this section, we shall prove that the proposed scheme can work correctly by the following four theorems.

Theorem 3.1 *In the proxy share generation phase, the designated clerk C_O can verify the validity of σ_{O_i} sent from U_{O_i} by Equation (5).*

Proof. From Equation (4), we have $\sigma_{O_i} = t_{O_i}K + x_{O_i}y_{O_i}h(M_w, K, AOSID)$ mod q. By raising both sides of the above equation to exponents with base g, we have

$$g^{\sigma_{O_i}} = g^{t_{O_i}K + x_{O_i}y_{O_i}h(M_w,K,AOSID)} \bmod p.$$

According to Equations (1) and (2), the above equation can be rewritten as

$$g^{\sigma_{O_i}} = k_{O_i}^{K} y_{O_i}^{y_{O_i}h(M_w,K,AOSID)} \bmod p.$$

QED.

Theorem 3.2 *If the proxy share is constructed correctly, it will pass the verification of Equation (7).*

Proof. From Equations (4) and (6), we have

$$\begin{aligned} \sigma_O &= \prod_{i=1}^{t_1'} \sigma_{O_i}, \\ &= \prod_{i=1}^{t_1'} t_{O_i}K + x_{O_i}y_{O_i}h(M_w, K, AOSID) \bmod q. \end{aligned}$$

By raising both sides of the above equation to exponents with base g, we have

$$g^{\sigma_O} = \prod_{i=1}^{t_1'} k_{O_i}^{K} y_{O_i}^{y_{O_i}h(M_w,K,AOSID)} \bmod p.$$

In accordance with Equation (3), the above equation can be rewritten as

$$g^{\sigma_O} = K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i}h(M_w,K,AOSID)} \bmod p.$$

QED.

Theorem 3.3 *In the proxy signature generation phase, the designated clerk C_P can verify any individual proxy signature s_{P_j} sent from U_{P_j} by Equation (10).*

Proof. Raising both sides of Equation (9) to exponents with base g, we have

$$\begin{aligned} g^{s_{P_j}} &= g^{t_{P_j}R+(\sigma_O t_2'^{-1}+x_{P_j}y_{P_j})h(M,R,APSID)}, \\ &= r_{P_j}^R (g^{\sigma_O t_2'^{-1}} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p. \end{aligned}$$

According to Equation (7), the above equation can be rewritten as

$$g^{s_{P_j}} = r_{P_j}^R ((K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2'^{-1}} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p.$$

QED.

Theorem 3.4 *If the proxy signature is constructed correctly, it will pass the verification of Equation (12).*

Proof. In Equations (9) and (11), we have

$$\begin{aligned} S &= \prod_{j=1}^{t_2'} s_{P_j}, \\ &= \prod_{j=1}^{t_2'} t_{P_j}R + (\sigma_O t_2'^{-1} + x_{P_j}y_{P_j})^{h(M,R,APSID)} \bmod q. \end{aligned}$$

By raising both sides of the above equation to exponents with base g, we have

$$g^S = \prod_{j=1}^{t_2'} r_{P_j}^R (g^{\sigma_O t_2'^{-1}} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p.$$

According to Equations (7) and (8), the above equation can be written as

$$g^S = R^R (K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \prod_{j=1}^{t_2'} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p.$$

QED.

4 Security Analysis

The security of the proposed scheme is based on the difficulty of breaking the one-way hash function [1] and the discrete logarithms [2, 5, 7]. In this section, we shall consider some possible attacks against the proposed scheme. We shall prove that the proposed scheme can successfully withstand these possible attacks.

Attack 1: An adversary tries to receive a signer's private key from all available public information.

Analysis of attack 1: Assume that the adversary wants to derive U_{O_i}'s private key x_{O_i} from Equation (1). It is as difficult as breaking the discrete logarithms to obtain U_{O_i}'s private key x_{O_i}. Similarly, the adversary will meet the same difficulty as she/he tries to obtain U_{P_j}'s private key x_{P_j}.

Attack 2: An adversary tries to forge a valid proxy share of a chosen M_w to pass the proxy share verification equation.

Analysis of attack 2: Let $Y_O = \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i}} \bmod p$. The adversary can rewrite Equation (7) as

$$g^{\sigma_O} = K^K Y_O^{h(M_w, K, AOSID)} \bmod p.$$

Y_O is a fixed value as the actual original signers' public keys are certified by CA. Given M'_w, K', $AOSID'$ and Y_O, it is difficult to determine σ'_O because of the difficulty of breaking the discrete logarithms. Again, given M'_w, $AOSID'$, σ'_O and Y_O, one can calculate a K' such that this equation holds. However, it is difficult to solve the one-way hash function and the discrete logarithms. Therefore, the proxy share verification equation is secure against the forgery attack.

Attack 3: An adversary tries to forge a valid proxy signature of some chosen M_w and M to pass the proxy signature verification equation.

Analysis of attack 3: Let V_O and V_P as follows:

$$V_O = K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod p, \tag{13}$$

$$V_P = \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}} \bmod p.$$

The attacker can rewrite Equation (12) as

$$g^S = R^R (V_O V_P)^{h(M,R,APSID)} \bmod p.$$

The value V_O depends on the parameters M_w, K and $AOSID$. V_P is a fixed value as the actual proxy signers' public keys are certified by CA. Given M', $APSID'$, V'_O and V'_P, it is difficult to determine R' and S' because of the difficulty of solving the one-way hash function and the discrete logarithms. Again, given M', $APSID'$, R', S' and V'_P, one can calculate a V'_O such that this equation holds. However, it is difficult to find M'_w, $AOSID'$ and K' such that Equation (13) holds. The difficulty here is also based on the one-way hash function and the discrete logarithms. Therefore, the proxy signature verification equation is secure against the forgery attack.

Attack 4: A malicious original signer U_{O_k} ($U_{O_k} \in G_O$), without any private keys of the other original signers, attempts to forge a valid proxy share for an arbitrary warrant in an insider forgery attack [12].

Analysis of attack 4: To forge the proxy share, U_{O_k} has to change her/his public key after the public keys of the other $t_1 - 1$ or more original signers have been determined. Assume that U_{O_k} waits until she/he derives any $t_1 - 1$ original signers' public keys y_{O_i}. She/he substitutes her/his public key y_{O_k}.

U_{O_k} chooses a random integer x'_{O_k} as her/his private key. Then, U_{O_k} has to make her/his public key y'_{O_k} to satisfy the following equation:

$$y_{O_k}^{\prime y'_{O_k}} = g^{x'_{O_k}} \left(\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}}\right)^{-1} \bmod p. \quad (14)$$

Next, U_{O_k} can choose a random integer t_{O_k} and compute K and σ_O as follows:

$$K = g^{t_{O_k}} \bmod p,$$

$$\sigma_O = t_{O_k} K + x'_{O_k} h(M_w, K, AOSID) \bmod q.$$

Thus, σ_O is a valid proxy share. This is because

$$\begin{aligned}
g^{\sigma_O} &= K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}, \\
&= K^K \big(y'^{\,y'_{O_k}}_{O_k} \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}}\big)^{h(M_w, K, AOSID)}, \\
&= K^K \big(g^{x'_{O_k}} \big(\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}}\big)^{-1} \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}}\big)^{h(M_w, K, AOSID)}, \\
&= g^{t_{O_k} K + x'_{O_k} h(M_w, K, AOSID)} \bmod p.
\end{aligned}$$

However, U_{O_k} cannot succeed in creating the proxy share σ_O. In Equation (14), suppose U_{O_k} determines the value x'_{O_k} first. She/he has to obtain the value y'_{O_k} by solving the difficult problem. On the other hand, if U_{O_k} wants to fix y'_{O_k} first, then she/he has to solve the discrete logarithms to find her/his private key x'_{O_k}. Therefore, the malicious original signer cannot successfully forge any proxy share for any warrant by launching the insider forgery attack.

Attack 5: A malicious original signer U_{O_k} ($U_{O_k} \in G_O$), without any private keys of the other original signers and proxy signers, attempts to forge a valid proxy signature for an arbitrary message in an insider forgery attack.

Analysis of attack 5: As with Attack 4, U_{O_k} will have to face the difficult problem of generating a valid proxy signature. U_{O_k} waits until she/he obtains any $t_1 - 1$ original signers' public key y_{O_i} and any t_2 proxy signers' public keys y_{P_j}. U_{O_k} modifies her/his public key y_{O_k}. If U_{O_k} also chooses a random integer x'_{O_k} as her/his private key, then U_{O_k} has to make her/his public key y'_{O_k} to satisfy the following equation:

$$y'^{\,y'_{O_k}}_{O_k} = g^{x'_{O_k}} K^{-K h(M_w, K, AOSID)^{-1}} \big(\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}}\big)^{-1} \big(\prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}}\big)^{-h(M_w, K, AOSID)^{-1}} \bmod p. \quad (15)$$

After that, U_{O_k} selects a random integer t_{P_k} and calculates R and S as follows:

$$\begin{aligned}
R &= g^{t_{P_k}} \bmod p, \\
S &= t_{P_k} R + x'_{O_k} h(M_w, K, AOSID) h(M, R, APSID) \bmod q.
\end{aligned}$$

Thus, (M_w, K, $AOSID$, M, R, S, $APSID$) is a valid proxy signature. It is because the forged proxy signature can satisfy Equation (12) as follows:

$$
\begin{aligned}
g^S &= R^R(K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i}h(M_w,K,AOSID)} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R(K^K (y_{O_k}^{\prime y_{O_k}} \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{h(M_w,K,AOSID)} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R(K^K (g^{x'_{O_k}} K^{-Kh(M_w,K,AOSID)^{-1}} (\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{-1} (\prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{-h(M_w,K,AOSID)^{-1}} \times \\
&\quad \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{h(M_w,K,AOSID)} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R(g^{x'_{O_k}h(M_w,K,AOSID)})^{h(M,R,APSID)}, \\
&= g^{t_{P_k}R+x'_{O_k}h(M_w,K,AOSID)h(M,R,APSID)} \bmod p.
\end{aligned}
$$

Nevertheless, U_{O_k} still cannot obtain a valid proxy signature. According to Equation (15), if U_{O_k} first determines the integer x'_{O_k}, she/he has to obtain the value of y'_{O_k} by solving the difficult problem. On the other hand, if U_{O_k} fixes the integer y'_{O_k}, she/he will have to solve the discrete logarithms to find the value of x'_{O_k}. Therefore, the insider forgery attack launched by a malicious original signer in the original group cannot work.

Attack 6: The malicious proxy signer U_{P_k} ($U_{P_k} \in G_P$), without any private keys of the other proxy signers, attempts to forge a valid proxy signature for an arbitrary message in an insider forgery attack.

Analysis of attack 6: As with Attacks 4 and 5, U_{P_k} cannot succeed in creating the proxy signature. Without loss of generality, suppose that U_{P_k} wants to update her/his public key y_{P_k}. U_{P_k} also waits until she/he obtains any $t_2 - 1$ proxy signers' public keys y_{P_j}. Then, she/he changes her/his public key y_{P_k}. First, U_{P_k} chooses a random integer x'_{P_k} and makes her/his public key y'_{P_k} as follows:

$$
y_{P_k}^{\prime y'_{P_k}} = g^{x'_{P_k}} (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{-1} \bmod p. \quad (16)
$$

Later, U_{P_k} selects a random integer t_{P_k} and computes R and S as follows:

$$
\begin{aligned}
R &= g^{t_{P_k}} \bmod p, \\
S &= t_{P_k} R + (\sigma_O + x'_{P_k}) h(M, R, APSID) \bmod q.
\end{aligned}
$$

Thus, (M_w, K, $AOSID$, M, R, S, $APSID$) is a valid proxy signature. It is because the forged proxy signature can satisfy Equation (12) as follows:

$$
\begin{aligned}
g^S &= R^R (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} y_{P_k}^{\prime y'_{P_k}} \prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} g^{x'_{P_k}} (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{-1} \prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= R^R (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} g^{x'_{P_k}})^{h(M,R,APSID)}, \\
&= g^{t_{P_k} R + (\sigma_O + x'_{P_k}) h(M,R,APSID)} \bmod p.
\end{aligned}
$$

However, U_{P_k} cannot create a valid proxy signature. In Equation (16), assume that the value of x'_{P_k} is determined. It is an extremely difficult problem to find a y'_{P_k} satisfying the equation. On the other hand, if U_{P_k} determines the integer y'_{P_k} first, then she/he has to solve the discrete logarithms to find the value of x'_{P_k}. Hence, the insider forgery attack launched by a malicious proxy signer cannot work.

Attack 7: The malicious original signer U_{O_k} ($U_{O_k} \in G_O$) tries to impersonate any $t_1 - 1$ or more original signers in G_O and to forge the proxy share without the agreement of these original signers in a direct forgery attack.

Analysis of attack 7: Suppose U_{O_k} waits until she/he obtains any $t_1 - 1$ original signers' public keys. Then, U_{O_k} chooses a random integer t_{O_k} and an arbitrary M_w, and calculates K and σ_O as follows:

$$
\begin{aligned}
K^K &= g^{t_{O_k}} (\prod_{i=1}^{t-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)})^{-1} \bmod p, \qquad (17) \\
\sigma_O &= t_{O_k} + x_{O_k} y_{O_k} h(M_w, K, AOSID) \bmod q.
\end{aligned}
$$

Thus, σ_O is a valid proxy sharing because the forged proxy share can pass Equation (7) as follows:

$$\begin{aligned}
g^{\sigma_O} &= K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}, \\
&= g^{t_{O_k}} (\prod_{i=1}^{t-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)})^{-1} y_{O_k}^{y_{O_k} h(M_w, K, AOSID)} \prod_{i=1}^{t-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}, \\
&= g^{t_{O_k} + x_{O_k} y_{O_k} h(M_w, K, AOSID)} \bmod p.
\end{aligned}$$

However, U_{O_k} cannot succeed in generating the proxy share σ_O. It is an extremely difficult problem to find a K satisfying Equation (17). Suppose U_{O_k} determines the integer K first. In this case, she/he has to solve the discrete logarithms to find the value of t_{O_k}. Thus, the direct forgery attack launched by the malicious original signer in the original group cannot succeed.

Attack 8: A malicious proxy signer U_{P_k} ($U_{P_k} \in G_P$) tries to impersonate any $t_2 - 1$ or more proxy signers in G_P and to forge the proxy signature without the agreement of these proxy signers.

Analysis of attack 8: As with Attack 7, U_{P_k} will meet the intractability problem. Suppose that U_{P_k} waits until she/he obtains any $t_2 - 1$ proxy signers' public keys. Then, U_{P_k} chooses a random integer t_{P_k} and calculates R and S as follows:

$$\begin{aligned}
R^R &= g^{t_{P_k}} (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j} h(M, R, APSID)})^{-1} \bmod p, \\
S &= t_{P_k} + (\sigma_O + x_{P_k} y_{P_k}) h(M, R, APSID) \bmod q.
\end{aligned} \tag{18}$$

Thus, (M_w, K, $AOSID$, M, R, S, $APSID$) is a valid proxy signature because the forged proxy signature can satisfy Equation (12) as follows:

$$\begin{aligned}
g^S &= R^R (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M, R, APSID)}, \\
&= g^{t_{P_k}} (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j} h(M, R, APSID)})^{-1} (K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \times \\
&\quad y_{P_k}^{y_{P_k}} \prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{h(M, R, APSID)},
\end{aligned}$$

$$
\begin{aligned}
&= g^{t_{P_k}}(K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} y_{P_k}^{y_{P_k}})^{h(M,R,APSID)}, \\
&= g^{t_{P_k}+(\sigma_O + x_{P_k} y_{P_k}) h(M,R,APSID)} \bmod p.
\end{aligned}
$$

However, U_{P_k} still cannot create a valid proxy signature. It is an extremely difficult problem to find R satisfying Equation (18). If U_{P_k} determines the integer R first, she/he has to solve the discrete logarithms to find the value of t_{P_k}. Thus, the direct forgery attack launched by the malicious proxy signer cannot work.

5 Special Cases of the Proposed Scheme

In this section, we shall discuss some special cases of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme. They are the $(1-t_2/n_2)$ proxy signature scheme, the (t_1/n_1-1) proxy signature scheme, and the $(1-1)$ proxy signature scheme.

5.1 The $(1 - t_2/n_2)$ Proxy Signature Scheme

In the $(1 - t_2/n_2)$ proxy signature scheme, any t_2 or more proxy signers from a designated proxy group of n_2 proxy signers can sign messages on behalf of the original signer. The major difference here is in the proxy share generation phase.

The system parameters are the same as those defined in Section 2. Let U_O be the original signer. In the proxy share generation phase, U_O delegates the signing capability to G_P. First, U_O selects a random integer $t \in Z_q^*$ and calculates K and σ_O as follows:

$$
\begin{aligned}
K &= g^t \bmod p, \\
\sigma_O &= tK + x_O y_O h(M_w, K) \bmod q.
\end{aligned}
$$

Then, the original signer broadcasts (M_w, K, σ_O) to G_P. After receiving (M_w, K, σ_O), each $U_{P_j} \in G_P$ checks whether the following equation holds:

$$
g^{\sigma_O} = K^K y_O^{y_O h(M_w, K)} \bmod p. \tag{19}
$$

If it does, each U_{P_j} uses σ_O as her/his proxy share.

In the proxy signature generation phase, any t_2 or more proxy signers, as a group, can sign a message M cooperatively on behalf of G_P. The procedure is similar to that in Section 2. Each U_{P_j} calculates R and s_{P_j} in Equations (8) and (9) and sends s_{P_j} to the designated clerk C_P. C_P can verify the validity of s_{P_j} according to the following equation:

$$g^{s_{P_j}} = r_{P_j}^R((K^K y_O^{y_O h(M_w,K)})^{t_2'^{-1}} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p.$$

If the equation holds, then C_P derives S from Equation (11), and the proxy signature on message M is $(M_w, K, M, R, S, APSID)$.

In the proxy signature verification phase, the verifier checks the validity of the proxy signature and identifies the actual proxy signers from the proxy group through the following equation:

$$g^S = R^R(K^K y_O^{y_O h(M_w,K)} \prod_{j=1}^{t_2'} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod p.$$

If the above equation holds, the verifier can make sure of the validity of the proxy signature and identify the actual proxy signers from $APSID$.

5.2 The $(t_1/n_1 - 1)$ Proxy Signature Scheme

In the $(t_1/n_1 - 1)$ proxy signature scheme, a designated proxy signer is allowed to generate a proxy signature on behalf of the original signer group. Any t_1 or more of the original signers can cooperatively delegate their signing capability to the designated proxy signer.

The system parameters are the same as those defined in Section 2. Let U_P be the designated proxy signer. The major difference here is in the proxy signature generation phase.

In the proxy share generation phase, the steps are the same as those in Section 2. D_O broadcasts $(M_w, K, \sigma_O, AOSID)$ to U_P. After receiving $(M_w, K, \sigma_O, AOSID)$, U_P checks whether Equation (7) holds. If it does, U_P confirms the validity of $(M_w, K, \sigma_O, AOSID)$.

In the proxy signature generation phase, U_P is allowed to generate a proxy signature for the message M on behalf of G_O. First, U_P randomly selects an integer t and calculates R and S as follows:

$$\begin{aligned} R &= g^t \bmod p, \\ S &= tR + (\sigma_O + x_P y_P) h(M, R) \bmod q. \end{aligned}$$

Then, the proxy signature of M is $(M_w, K, AOSID, M, R, S)$.

Finally, the verifier checks the validity of the proxy signature and identifies the actual original signers from the original group through the following equation:

$$g^S = R^R (K^K \prod_{i=1}^{t'} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} y_P^{y_P})^{h(M,R)} \bmod p.$$

If the above equation holds, the verifier can believe in the validity of the proxy signature and know the identities of the actual original signers from $AOSID$.

5.3 The $(1-1)$ Proxy Signature Scheme

The $(1-1)$ proxy signature scheme is the basic proxy signature scheme. The original signer delegates her/his signing capability to a designated proxy signer. The proxy signer generates a proxy signature on behalf of the original signer.

The system parameters are also the same as those defined in Section 2. Let U_O be the original signer and U_P be the proxy signer. The steps of the proxy share generation phase are the same as those in Section 5.1. The original signer gives (M_w, K, σ_O) to a designated proxy signer. After confirming the validity of (M_w, K, σ_O) by checking Equation (19), U_P performs the same steps as those taken in the proxy signature generation phase in Section 5.2. Then, the proxy signature of message M is (M_w, K, M, R, S). Finally, the verifier confirms the validity of the proxy signature by checking the following equation:

$$g^S = R^R (K^K y_O^{y_O h(M_w, K)} y_P^{y_P})^{h(M,R)} \bmod p.$$

If the equation holds, the proxy signature (M_w, K, M, R, S) of M is valid.

The security of the $(t_1/n_1 - 1)$, $(1 - t_2/n_2)$ and $(1 - 1)$ proxy signature schemes presented above is also guaranteed by the analysis in Section 4. Those attacks will not succeed here because of the difficulty of breaking the one-way hash function and the discrete logarithms.

6 Conclusion

In this article, we have proposed a new generalized of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on discrete logarithms. This scheme can be applied to every proxy situation, namely the $(t_1/n_1 - 1)$ proxy signature, the $(1 - t_2/n_2)$ proxy signature, and the $(1 - 1)$ proxy signature. For avoiding the abuse of the signing capability, the proposed scheme provides the ability to identify the actual original signers and the actual proxy signers. The actual original signers cannot deny delegating the warrant, and the actual proxy signers cannot deny signing the proxy signatures either. Some possible attacks such as equation attacks, insider forgery and direct forgery attacks have been considered. None of them can successfully break the proposed scheme.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[2] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[3] Chien-Lung Hsu, Tzong-Sun Wu, and Tzong-Chen Wu, "New nonrepudiable threshold proxy signature scheme with known signers," *The Journal of Systems and Software*, no. 58, pp. 119–124, 2001.

[4] 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, no. 2, pp. 1–8, 2000.

[5] Min-Shiang Hwang, Chin-Chen Chang, and Kuo-Feng Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[6] Min-Shiang Hwang, Cheng-Chi Lee, and Yan-Chi Lai, "Traceability on low-computation partially blind signatures for electronic cash," *IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences*, vol. E85-A, no. 5, pp. 1181–1182, 2002.

[7] Min-Shiang Hwang, Cheng-Chi Lee, and Eric Jui-Lin Lu, "Cryptanalysis of the batch verifying multiple DSA-type digital signatures," *Pakistan Journal of Applied Sciences*, vol. 1, no. 3, pp. 287–288, 2001.

[8] Min-Shiang Hwang, Iuon-Chung Lin, and Kuo-Feng Hwang, "Cryptanalysis of the batch verifying multiple RSA digital signatures," *Informatica*, vol. 11, no. 1, pp. 15–19, 2000.

[9] Min-Shiang Hwang, Eric Jui-Lin Lu, and Iuon-Chang Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions on Knowledge and Data Engineering*, accepted (Aug. 22, 2002) and to appear.

[10] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," *Proc. of ICICS'97, LNCS 1334*, pp. 223–232, 1997.

[11] N.Y. Lee, T. Hwang, and C.H. Wang, "On Zhang's nonrepudiable proxy signature schemes," *ACISP'98, LNCS 1438*, pp. 415–422, Jul. 1998.

[12] Z. C. Li, L. C. K. Hui, K. P. Chow, C. F. Chong, H. H. Tsang, and H. W. Chan, "Cryptanalysis of Harn digital multisignature scheme with distinguished signing authorities," *Electronics Letters*, vol. 36, no. 4, p-p. 314–315, 2000.

[13] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign message," *IEICE Trans. Fundamentals*, vol. E79-A, pp. 1338–1353, Sep. 1996.

[14] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[15] H. M. Sun, "An efficient nonrepudiable threshold proxy signature scheme with known signers," *Computer Communications*, vol. 22, no. 8, pp. 717–722, 1999.

[16] Shiang-Feng Tzeng, Chen-Ying Yang, and Min-Shiang Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Proceeding of* 12^{th} *National Conference on Information Security, R.O.C.*, pp. 285–292, 2002.

[17] L. Yi, G. Bai, and G. Xiao, "Proxy multi-signature scheme: A new type of proxy signature scheme," *Electronics Letters*, vol. 36, no. 6, pp. 527–528, 2000.

[18] K. Zhang, "Threshold proxy signature schemes," *1997 Information Security Workshop*, pp. 191–197, 1997.