ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Computer Standards & Interfaces 2227 (2003) xxx–xxx

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Generalization of proxy signature based on elliptic curves

Min-Shiang Hwang[a,*], Shiang-Feng Tzeng[b], Chwei-Shyong Tsai[c]

[a] *Graduate Institute of Networking and Communication Engineering, Chaoyang University of Technology, 168 Gifeng E. Road, Wufeng, Taichung County 413, Taiwan, ROC*

[b] *Department of Information Management, Chaoyang University of Technology, 168 Gifeng E. Road, Wufeng, Taichung County 413, Taiwan, ROC*

[c] *Department of Information Management, National Taichung Institute of Technology, 129 Sec. 3, San-min Road, Taichung 404, Taiwan, ROC*

Received 12 June 2003; received in revised form 12 June 2003; accepted 15 June 2003

Abstract

In the past, proxy signature schemes were usually proposed to deal with one or two separate proxy conditions each. To make a difference, in this paper, the authors shall present a generalized version of proxy signature scheme. Compared to earlier proxy signature schemes, our novel scheme can be applied to every possible proxy situation. The proposed 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 a 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. With this article, we also aim to demonstrate how to specify proxy signature schemes on elliptic curve over finite fields. In addition, some possible attacks have also been considered, and our security analysis will show that none of them can successfully break the proposed scheme.

Keywords: Cryptography; Digital signature; Elliptic curves; Proxy signature; Threshold proxy signature; Proxy multi-signature scheme

1. Introduction

In an organization, suppose a manager needs to go on a business trip. She/he has to find a proxy person to deal with her/his work at the office. The manager can delegate her/his signing capability to a designated proxy signer so that the designated proxy signer can generate a signature on behalf of the manager. Common digital signature schemes [3,7,9,10,18], however, are not applicable in the above general situation. In order 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. Up to the present time, there have been quite a number of proxy signature schemes [4,5,11–13,20–22] proposed.

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 into 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.

* Corresponding author. Tel.: +886-4-23323000x7241, fax: +886-4-23742337.

E-mail address: mshwang@mail.cyut.edu.tw (M.-S. Hwang).

0920-5489/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S0920-5489(03)00075-8

Recently, $(1 - t_2/n_2)$ proxy signature schemes have drawn much attention [4,5,11–13,22]. In such a design, the original signer is able to delegate her/his signing capability to the proxy group of n_2 proxy signers. Any t_2 or more proxy signers in the proxy group can cooperatively create the proxy signature.

On the other hand, a $(n_1 - 1)$ proxy signature scheme can be found in [Ref. 21]. 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 generate 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 Ref. [19], 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 Refs. [4,5] 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. 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.

Since the first public key cryptosystem was proposed, several kinds of cryptographic assumptions have been made for the development of both security and efficiency. At present, only the discrete logarithm problem and factorization problem of digital signatures, such as DSA [17] and RSA [1,8,10,18], are widely accepted. The elliptic curve cryptosystem (ECC) [16], on the other hand, is constructed by integer points over elliptic curves in finite fields. The ECC can reach the same level of security constituted by DSA or RSA but provides greater efficiency than either discrete logarithm or factorization systems. The aim of this article is to complete the description of our generalized proxy signature scheme on an elliptic curve.

The remainder of this article is organized as follows. In Section 2, our novel $(t_1/n_1 - t_2/n_2)$ proxy signature scheme will be described in detail. In Section 3, we shall prove that the proposed scheme can work correctly. In Section 4, some special cases of the proposed scheme will be presented. After that, we shall analyze the security of the proposed scheme 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 service provided by C_O for G_O is 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 service provided by C_P for G_P is to collect the individual proxy signatures generated by the proxy signers so as to construct the final proxy signature. Besides, C_P can check 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 $(1 \le t_1 \le n_1)$ can represent G_O to delegate the signing capability. Similarly, any t_2 or more out of n_2 proxy signers $(1 \le t_2 \le n_2)$ 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 as follows:

- E: an elliptic curve over a finite field $GF(q)$.
- G: a base point on E.

- n: the order of the base point.
- $h(\cdot)$: 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), (t_2, n_2), and the valid delegation time, etc.
- AOSID: (Actual Original Signers' ID) the identities of the actual original signers.
- APSID: (Actual Proxy Signers' ID) the identities of the actual proxy signers.

Each user U_i has a randomly selected private key x_i, where $1 \le x_i \le n-1$, and a public key y_i certified by the certificate authority (CA) as follows:

$$y_i = x_i G. \tag{1}$$

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

The proposed scheme can be divided into the following 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 $D_O=\{U_{O_1}, U_{O_2},\ldots, U_{O_{t_1}}\}$ be the actual original signers, where $t_1 \le t_1' \le n_1$. D_O be as a group executes the following steps to delegate the signing capability to G_P.

(1) Select a random integer t_{O_i}, where $1 \le t_{O_i} \le n-1$, and calculate k_{O_i} as follows:

$$k_{O_i} = t_{O_i} G. \tag{2}$$

Next, broadcast k_{O_i} to the other $t_1'-1$ original signers in D_O and the designated clerk C_O.

(2) After receiving k_{O_j} $(j=1, 2,\ldots,t_1'; j\neq i)$, each $U_{O_i}\in D_O$ calculates K and σ_{O_i} as follows:

$$K = \sum_{i=1}^{t_1'} k_{O_i}, \tag{3}$$

$$\sigma_{O_i} = (K)_x t_{O_i} + h(M_w,\ (K)_x,\ \mathrm{AOSID})(y_{O_i})_x x_{O_i} \bmod n. \tag{4}$$

Here, $(\cdot)_x$ denotes the x-coordinate of point $(\cdot)$ on E.

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

(4) For each received σ_{O_i} $(i=1, 2,\ldots, t_1')$, C_O checks whether the following equation holds:

$$\sigma_{O_i} G = (K)_x k_{O_i} + h(M_w,\ (K)_x,\ \mathrm{AOSID})(y_{O_i})_x y_{O_i}. \tag{5}$$

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

$$\sigma_O = \sum_{i=1}^{t_1'} \sigma_{O_i} \bmod n. \tag{6}$$

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

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

$$\sigma_O G = (K)_x K + h(M_w,\ (K)_x,\ \mathrm{AOSID}) \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i}. \tag{7}$$

If it does, each $U_{P_j}\in G_P$ confirms the validity of $(M_w, K, \sigma_O, \mathrm{AOSID})$. 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 $D_P=\{U_{P_1}, U_{P_2},\ldots, U_{P_{t_2}}\}$ be the actual proxy signers for $t_2 \le t_2' \le n_2$. D_P be as a group executes the following steps to generate a proxy signature.

(1) Select a random integer t_{P_j}, where $1 \le t_{P_j} \le n-1$, and calculate r_{P_j} as follows:

$$r_{P_j} = t_{P_j} G.$$

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} ($k=1, 2,\ldots, t_2'$, $k\neq j$), each $U_{P_j}\in D_P$ calculates R and s_{P_j} as follows:

$$R=\sum_{j=1}^{t_2'} r_{P_j}, \tag{8}$$

$$s_{P_j}=(R)_x t_{P_j}+h(M,\ (R)_x,\ \mathrm{APSID})\times (t_2'^{-1}\sigma_O+(y_{P_j})_x x_{P_j})\bmod n. \tag{9}$$

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

(3) For each received s_{P_j} ($j=1, 2,\ldots, t_2'$), C_P checks whether the following equation holds:

$$s_{P_j}G=(R)_x r_{P_j}+h(M,\ (R)_x,\ \mathrm{APSID})\times\left(t_2'^{-1}\left((K)_x K+h(M_w,\ (K)_x,\ \mathrm{AOSID})\times\sum_{i=1}^{t_1'}(y_{O_i})_x y_{O_i}\right)+(y_{P_j})_x y_{P_j}\right). \tag{10}$$

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

$$S=\sum_{j=1}^{t_2'} s_{P_j} \bmod n. \tag{11}$$

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 who 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:

$$SG=(R)_x R+h(M,\ (R)_x,\ \mathrm{APSID})\times\left((K)_x K+h(M_w,\ (K)_x,\ \mathrm{AOSID})\times\sum_{i=1}^{t_1'}(y_{O_i})_x y_{O_i}+\sum_{j=1}^{t_2'}(y_{P_j})_x y_{P_j}\right). \tag{12}$$

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 checking 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 Eq. (5).*

Proof. By raising both sides of Eq. (4) by multiplying them by the base point G, we have

$$\sigma_{O_i}G=((K)_x t_{O_i}+h(M_w,(K)_x,\ \mathrm{AOSID})(y_{O_i})_x x_{O_i})G.$$

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

$$\sigma_{O_i}G=(K)_x k_{O_i}+h(M_w,(K)_x,\ \mathrm{AOSID})(y_{O_i})_x y_{O_i}.\ \square$$

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

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

$$\sigma_O=\sum_{i=1}^{t_1'}\sigma_{O_i},$$

$$=\sum_{i=1}^{t_1'}(K)_x t_{O_i}+h(M_w,(K)_x,\ \mathrm{AOSID})\times(y_{O_i})_x x_{O_i}\bmod n.$$

By raising both sides of the above equation by multiplying them by the base point G, we have

$$\sigma_O G=\sum_{i=1}^{t_1'}(K)_x k_{O_i}+h(M_w,(K)_x,\ \mathrm{AOSID})(y_{O_i})_x y_{O_i}.$$

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

$$\sigma_O G=(K)_x K+h(M_w,(K)_x,\ \mathrm{AOSID})\sum_{i=1}^{t_1'}(y_{O_i})_x y_{O_i}.$$

$\square$

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 checking Eq. (10).*

Proof. Raising both sides of Eq. (9) by multiplying them by the base point G, we have

$$\begin{aligned} s_{\mathrm{P}_j}G &= ((R)_x t_{\mathrm{P}_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times (t_2'^{-1}\sigma_{\mathrm{O}} + (y_{\mathrm{P}_j})_x x_{\mathrm{P}_j}))G, \\ &= (R)_x r_{\mathrm{P}_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times (t_2'^{-1}\sigma_{\mathrm{O}}G + (y_{\mathrm{P}_j})_x y_{\mathrm{P}_j}). \end{aligned}$$

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

$$\begin{aligned} s_{\mathrm{P}_j}G &= (R)_x r_{\mathrm{P}_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times \left(t_2'^{-1} \left((K)_x K + h(M_{\mathrm{w}}, (K)_x,\ \mathrm{AOSID}) \right.\right. \\ &\quad \left.\left. \times \sum_{i=1}^{t_1'} (y_{\mathrm{O}_i})_x y_{\mathrm{O}_i} \right) + (y_{\mathrm{P}_j})_x y_{\mathrm{P}_j} \right). \end{aligned}$$

□

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

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

$$\begin{aligned} S &= \sum_{j=1}^{t_2'} s_{\mathrm{P}_j}, \\ &= \sum_{j=1}^{t_2'} (R)_x t_{\mathrm{P}_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times (t_2'^{-1}\sigma_{\mathrm{O}} + (y_{\mathrm{P}_j})_x x_{\mathrm{P}_j}) \bmod n. \end{aligned}$$

By raising both sides of the above equation by multiplying them by the base point G, we have

$$\begin{aligned} SG &= \sum_{j=1}^{t_2'} (R)_x r_{\mathrm{P}_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times (t_2'^{-1}\sigma_{\mathrm{O}}G + (y_{\mathrm{P}_j})_x y_{\mathrm{P}_j}). \end{aligned}$$

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

$$\begin{aligned} SG &= (R)_x R + h(M,\ (R)_x,\ \mathrm{APSID}) \\ &\quad \times \left((K)_x K + h(M_{\mathrm{w}}, (K)_x,\ \mathrm{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{\mathrm{O}_i})_x y_{\mathrm{O}_i} + \sum_{j=1}^{t_2'} (y_{\mathrm{P}_j})_x y_{\mathrm{P}_j} \right). \end{aligned}$$

□

4. 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.

4.1. 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, where $1 \le t \le n - 1$, and calculates K and σ_{O} as follows:

$$K = tG,$$

$$\sigma_{\mathrm{O}} = (K)_x t + h(M_{\mathrm{w}},\ (K)_x)(y_{\mathrm{O}})_x x_{\mathrm{O}} \bmod n.$$

Then, the original signer broadcasts (M_{w}, K, σ_{O}) to G_{P}. After receiving (M_{w}, K, σ_{O}), each $U_{\mathrm{P}_j} \in G_{\mathrm{P}}$ checks whether the following equation holds:

$$\sigma_{\mathrm{O}}G = (K)_x K + h(M_{\mathrm{w}},\ (K)_x)(y_{\mathrm{O}})_x y_{\mathrm{O}}. \tag{13}$$

If it does, each U_{P_j} as σ_{O} be 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 Eqs. (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:

$$s_{P_j}G = (R)_x r_{P_j} + h(M,\ (R)_x,\ \mathrm{APSID}) \times (t_2')^{-1}((K)_x K + h(M_w,\ (K)_x)(y_O)_x y_O) + (y_{P_j})_x y_{P_j}).$$

If the equation holds, then C_P derives S from Eq. (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:

$$SG = (R)_x R + h(M,\ (R)_x,\ \mathrm{APSID}) \times \left((K)_x K + h(M_w, (K)_x)(y_O)_x y_O + \sum_{j=1}^{t_2'} (y_{P_j})_x y_{P_j} \right).$$

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.

4.2. 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, σ_O, AOSID) to U_P. After receiving M_w, K, σ_O, AOSID), U_P checks whether Eq. (7) holds. If it does, U_P confirms the validity of (M_w, K, σ_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 from the interval $1 \le t \le n-1$ and calculates R and S as follows:

$$R = tG,$$

$$S = (R)_x t + h(M,\ (R)_x)(\sigma_O + (y_P)_x x_P) \bmod n.$$

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:

$$SG = (R)_x R + h(M,\ (R)_x) \times \left((K)_x K + h(M_w,\ (K)_x,\ \mathrm{AOSID}) \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + (y_P)_x y_P \right).$$

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

4.3. The (1–1) proxy signature scheme

The $(1-1)$ proxy signature scheme is the basic form of a proxy signature scheme where 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 4.1. 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 Eq. (13), U_P performs the same steps as those taken in the proxy signature generation phase in Section 4.2. 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:

$$SG = (R)_x R + h(M,\ (R)_x) \times ((K)_x K + h(M_w,\ (K)_x)(y_O)_x y_O + (y_P)_x y_P).$$

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

5. Security analysis

Let us discuss the security of the proposed scheme. Basically, the security of the proposed schemes is based on the difficulty of breaking the one-way hash function [2,6] and the elliptic curve discrete logarithm problem (ECDLP) [15]. Moreover, the security of the special cases of the proposed scheme is similar to the generalized scheme. Therefore, we only discuss and consider the security problem about the generalized scheme. 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 derive 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 Eq. (1). It is as difficult as breaking the ECDLP to obtain U_{O_i}'s private key x_{O_i}. Similarly, the adversary will encounter 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: Given $Y_O = \sum_{i=1}^{t_1}(y_{O_i})_x y_{O_i}$. The adversary can rewrite Eq. (7) as

$$\sigma_O G = (K)_x K + h(M_w,\ (K)_x,\ \mathrm{AOSID}) Y_O.$$

Y_O is a fixed value as the actual original signers' public keys have been 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 ECDLP. 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 ECDLP. 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: Given V_O and V_P as follows:

$$V_O = (K)_x K + h(M_w,\ (K)_x,\ \mathrm{AOSID}) \sum_{i=1}^{t_1}(y_{O_i})_x y_{O_i},$$

$$V_P = \sum_{j=1}^{t_2}(y_{P_j})_x y_{P_j}. \quad (14)$$

The attacker can rewrite Eq. (12) as

$$SG = (R)_x R + h(M,\ (R)_x,\ \mathrm{APSID})(V_O + V_P).$$

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 have been 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 ECDLP. 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 Eq. (14) holds. The difficulty here is also based on the one-way hash function and the ECDLP. Therefore, the proxy signature verification equation is secure against the forgery attack.

Attack 4: A malicious original signer U_{O_1} ($U_{O_1} \in G_O$), without any private keys of the other original signers, attempts to forge a valid proxy share for an arbitrary warrant to launch an insider forgery attack [14].

Analysis of Attack 4: To forge the proxy share, U_{O_1} 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_1} waits until she/he derives any $t_1 - 1$ original signers' public keys. She/he replaces her/his public key y_{O_1}.

U_{O_1} chooses a random integer x_{O_1}' as her/his private key. Then, U_{O_1} has to make her/his public key y_{O_1}' satisfy the following equation:

$$(y_{O_1}')_x y_{O_1}' = x_{O_1}' G - \sum_{i=2}^{t_1}(y_{O_i})_x y_{O_i}. \quad (15)$$

Next, U_{O_1} can choose a random integer t_{O_1}, where $1 \le t_{O_1} \le n-1$, and compute K and σ_O as follows:

$$K = t_{O_1} G,$$

$$\sigma_O = (K)_x t_{O_1} + h(M_w,\ (K)_x,\ \mathrm{AOSID}) x_{O_1}' \bmod n.$$

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

$$\begin{aligned}\sigma_O G &= (K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \sum_{i=1}^{t_1} (y_{O_i})_x y_{O_i},\\ &= (K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})\\ &\quad \times \left((y'_{O_1})_x y'_{O_1} + \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} \right),\\ &= (K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})\\ &\quad \times \left(x'_{O_1} G - \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} + \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} \right),\\ &= (K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})(x'_{O_1} G),\\ &= ((K)_x t_{O_k} + h(M_w,\ (K)_x,\ \text{AOSID}) x'_{O_1}) G.\end{aligned}$$

However, U_{O_1} cannot succeed in creating the proxy share σ_O. In Eq. (15), suppose U_{O_1} determines the value x'_{O_1} first. She/he has to obtain the value y'_{O_1} by solving the difficult problem. On the other hand, if U_{O_1} wants to fix y'_{O_1} first, then she/he has to solve the ECDLP to find her/his private key x'_{O_1}. 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_1} ($U_{O_1} \in G_O$), without any private keys of the other original signers and proxy signers, attempts to forge a valid proxy signature for an arbitrary message by launching an insider forgery attack.

Analysis of Attack 5: As with Attack 4, U_{O_1} will have to face the difficult problem of generating a valid proxy signature. Assume U_{O_1} waits until she/he obtains any $t_1 - 1$ original signers' public key and any t_2 proxy signers' public keys. U_{O_1} modifies her/his public key y_{O_1}. If U_{O_1} also chooses a random integer x'_{O_1} as her/his private key, then U_{O_1} has to make her/his public key y'_{O_1} satisfy the following equation:

$$\begin{aligned}(y'_{O_1})_x y'_{O_1} &= x'_{O_1} G - h(M_w,\ (K)_x,\ \text{AOSID})^{-1} (K)_x K\\ &\quad - \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} - h(M_w,\ (K)_x,\ \text{AOSID})^{-1}\\ &\quad \times \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j}. \qquad (16)\end{aligned}$$

After that, U_{O_1} selects a random integer t_{P_1} from the interval $1 \le t_{P_1} \le n-1$ and calculates R and S as follows:

$$R = t_{P_1} G,$$

$$\begin{aligned}S &= (R)_x t_{P_1} + h(M_w,\ (K)_x,\ \text{AOSID})\\ &\quad \times h(M,\ (R)_x,\ \text{APSID}) x'_{O_1} \bmod n.\end{aligned}$$

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

$$\begin{aligned}SG &= (R)_x R + h(M,\ (R)_x,\ \text{APSID})\\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})\right.\\ &\quad \left. \times \sum_{i=1}^{t_1} (y_{O_i})_x y_{O_i} + \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right),\\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID})\\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})\right.\\ &\quad \left. \times \left((y'_{O_1})_x y'_{O_1} + \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} \right) + \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right),\\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID})\\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID})\right.\\ &\quad \times \left(\left(x_{O_1}{}' G - h(M_w,\ (K)_x,\ \text{AOSID})^{-1} (K)_x K\right.\right.\\ &\quad \left.\left. - \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} - h(M_w,\ (K)_x,\ \text{AOSID})^{-1}\right.\right.\\ &\quad \left.\left. \times \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right) + \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} \right) + \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j} \Bigg),\\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID})\\ &\quad \times h(M_w,\ (K)_x, \text{AOSID}) x'_{O_1} G,\\ &= ((R)_x t_{P_1} + h(M_w,\ (K)_x,\ \text{AOSID})\\ &\quad \times h(M,\ (R)_x, \text{APSID}) x'_{O_1}) G.\end{aligned}$$

Nevertheless, U_{O_1} still cannot obtain a valid proxy signature. According to Eq. (16), if U_{O_1} first deter-

mines the integer x'_{O_1}, she/he has to obtain the value of y'_{O_1} by solving the difficult problem. On the other hand, if U_{O_1} fixes the integer y'_{O_1}, she/he will have to solve the ECDLP to find the value of x'_{O_1}. Therefore, the insider forgery attack launched by a malicious original signer in the original group cannot work.

Attack 6: A malicious proxy signer U_{P_1} ($U_{P_1} \in G_P$), without any private keys of the other proxy signers, attempts to forge a valid proxy signature for an arbitrary message by launching an insider forgery attack.

Analysis of Attack 6: As with Attacks 4 and 5, U_{P_1} cannot succeed in creating the proxy signature. Without loss of generality, suppose that U_{P_1} wants to update her/his public key. U_{P_1} also waits until she/he obtains any $t_2 - 1$ proxy signers' public keys. Then, she/he changes her/his public key y_{P_1}. First, U_{P_k} chooses a random integer x'_{P_1} and makes her/his public key y'_{P_1} as follows:

$$(y'_{P_1})_x y'_{P_1} = x'_{P_1} G - \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j}. \quad (17)$$

Later, U_{P_1} selects a random integer t_{P_1} from the interval $1 \le t_{P_1} \le n-1$ and computes R and S as follows:

$$R = t_{P_1} G,$$

$$S = (R)_x t_{P_1} + h(M,\ (R)_x,\ \text{APSID})(\sigma_O + x'_{P_1}) \bmod n.$$

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

$$\begin{aligned} SG &= (R)_x R + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right), \\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + \left((y'_{P_1})_x y'_{P_1} + \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j} \right) \right) \\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + \left(x'_{P_1} G - \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j} \right. \\ &\quad \left. \left. + \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j} \right) \right), \\ &= (R)_x R + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + x'_{O_1} G \right), \\ &= ((R)_x t_{P_1} + h(M,\ (R)_x,\ \text{APSID})(\sigma_O + x'_{P_1}))G. \end{aligned}$$

However, U_{P_1} cannot create a valid proxy signature. In Eq. (17), assume that the value of x'_{P_1} is determined. It is an extremely difficult problem to find a y_{P_1} satisfying the equation. On the other hand, if U_{P_1} determines the integer $y_{P_1}{}'$ first, then she/he has to solve the ECDLP to find the value of $x_{P_1}{}'$. Hence, the insider forgery attack launched by a malicious proxy signer cannot work.

Attack 7: The malicious original signer U_{O_1} ($U_{O_1} \in G_O$) 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.

Analysis of Attack 7: Suppose U_{O_1} waits until she/he obtains any $t_1 - 1$ original signers' public keys. Then, U_{O_1} chooses a random integer t_{O_1} from the interval $1 \le t_{O_1} \le n-1$ and an arbitrary M_w, and calculates K and σ_O as follows:

$$(K)_x K = t_{O_1} G - h(M_w,\ (K)_x,\ \text{AOSID}) \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i},$$

$$\sigma_O = t_{O_1} + h(M_w,\ (K)_x,\ \text{AOSID})(y_{O_k})_x x_{O_k} \bmod n. \quad (18)$$

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

$$\sigma_O G = (K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \sum_{i=1}^{t_1} (y_{O_i})_x y_{O_i},$$

$$= t_{O_1} G - h(M_w,\ (K)_x,\ \text{AOSID}) \sum_{i=2}^{t_1} (y_{O_i})_x y_{O_i} + h(M_w,\ (K)_x,\ \text{AOSID}) \sum_{i=1}^{t_1} (y_{O_i})_x y_{O_i},$$

$$= (t_{O_1} + h(M_w,\ (K)_x,\ \text{AOSID})(y_{O_1})_x x_{O_1}) G.$$

However, U_{O_1} cannot succeed in generating the proxy share σ_O. It is an extremely difficult problem to find a K satisfying Eq. (18). Suppose U_{O_1} determines the integer K first. In this case, she/he has to solve the ECDLP to find the value of t_{O_1}. 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_1} ($U_{P_1} \in G_P$) 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_1} will have to encounter meet the intractability problem. Suppose that U_{P_1} waits until she/he obtains any $t_2 - 1$ proxy signers' public keys. Then, U_{P_1} chooses a random integer t_{P_1}, where $1 \le t_{P_1} \le n - 1$, and calculates R and S as follows:

$$(R)_x R = t_{P_1} G - h(M,\ (R)_x,\ \text{APSID}) \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j},$$

$$S = t_{P_1} + h(M,\ (R)_x,\ \text{APSID})(\sigma_O + (y_{P_1})_x x_{P_1}) \bmod n. \quad (19)$$

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

$$\begin{aligned} SG &= (R)_x R + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right), \\ &= t_{P_1} G - h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \sum_{j=2}^{t_2} (y_{P_j})_x y_{P_j} + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + \sum_{j=1}^{t_2} (y_{P_j})_x y_{P_j} \right), \\ &= t_{P_1} G + h(M,\ (R)_x,\ \text{APSID}) \\ &\quad \times \left((K)_x K + h(M_w,\ (K)_x,\ \text{AOSID}) \right. \\ &\quad \left. \times \sum_{i=1}^{t_1'} (y_{O_i})_x y_{O_i} + (y_{P_1})_x y_{P_1} \right), \\ &= (t_{P_1} + h(M,\ (R)_x,\ \text{APSID})(\sigma_O + (y_{P_1})_x x_{P_1})) G. \end{aligned}$$

However, U_{P_1} still cannot create a valid proxy signature. It is an extremely difficult problem to find an R satisfying Eq. (19). If U_{P_1} determines the integer R first, she/he has to solve the ECDLP to find the value of t_{P_1}. Thus, the direct forgery attack launched by the malicious proxy signer cannot work. 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 this section. Those attacks will not succeed in those special cases either because of the difficulty of breaking the one-way hash function and the ECDLP.

6. Conclusion

In this article, we have proposed a novel generalized version of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on the elliptic curve discrete logarithm problem. The 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.

Acknowledgements

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

References

[1] C.-C. Chang, M.-S. Hwang, Parallel computation of the generating keys for RSA cryptosystems, IEE Electronics Letters 32 (15) (1996) 1365–1366.
[2] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (1976 Nov.) 644–654.
[3] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory IT-31 (1985 July) 469–472.
[4] C.-L. Hsu, T.-S. Wu, T.-C. Wu, New nonrepudiable threshold proxy signature scheme with known signers, The Journal of Systems and Software (58) (2001) 119–124.
[5] M.S. Hwang, I.C. Lin, E.J.L. Lu, A secure nonrepudiable threshold proxy signature scheme with known signers, International Journal of Informatica 11 (2) (2000) 1–8.
[6] M.-S. Hwang, C.-C. Chang, K.-F. Hwang, A water-marking technique based on one-way hash functions, IEEE Transactions on Consumer Electronics 45 (2) (1999) 286–294.
[7] M.-S. Hwang, C.-C. Lee, Y.-C. Lai, Traceability on low-computation partially blind signatures for electronic cash, IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences E85-A (5) (2002) 1181–1182.
[8] M.-S. Hwang, C.-C. Lee, Y.-C. Lai, Traceability on RSA-based partially signature with low computation, Applied Mathematics and Computation (2002).
[9] M.-S. Hwang, C.-C. Lee, E.J.-L. Lu, Cryptanalysis of the batch verifying multiple DSA-type digital signatures, Pakistan Journal of Applied Sciences 1 (3) (2001) 287–288.
[10] M.-S. Hwang, I.-C. Lin, K.-F. Hwang, Cryptanalysis of the batch verifying multiple RSA digital signatures, Informatica 11 (1) (2000) 15–19.
[11] M.-S. Hwang, E.J.-L. Lu, I.-C. Lin, A practical (*t*, *n*) threshold proxy signature scheme based on the RSA cryptosystem, IEEE Transactions on Knowledge and Data Engineering (2002 Aug. 22) (in press).
[12] S. Kim, S. Park, D. Won, Proxy signatures, revisited, Proc. of ICICS'97, LNCS, vol. 1334, 1997, pp. 223–232.
[13] N.Y. Lee, T. Hwang, C.H. Wang, On Zhang's nonrepudiable proxy signature schemes, ACISP'98, LNCS, vol. 1438, 1998 July, pp. 415–422.
[14] Z.C. Li, L.C.K. Hui, K.P. Chow, C.F. Chong, H.H. Tsang, H.W. Chan, Cryptanalysis of Harn digital multisignature scheme with distinguished signing authorities, Electronics Letters 36 (4) (2000) 314–315.
[15] A.J. Menezes, T. Okamoto, S.A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Transactions on Information Theory 39 (5) (1993) 1639–1646.
[16] V.S. Miller, Use of elliptic curves in cryptography, Advances in Cryptology, CRYPTO'85, Lecture Notes in Computer Science, vol. 218, 1985, pp. 417–426.
[17] National Institute of Standards and Technology (NIST), The digital signature standard proposed by NIST, Communications of the ACM 35 (7) (1992) 36–40.
[18] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM 21 (1978 Feb.) 120–126.
[19] H.M. Sun, An efficient nonrepudiable threshold proxy signature scheme with known signers, Computer Communications 22 (8) (1999) 717–722.
[20] S.-F. Tzeng, C.-Y. Yang, M.-S. Hwang, A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification, Proceeding of 12th National Conference on Information Security, R.O.C., 2002, pp. 285–292.
[21] L. Yi, G. Bai, G. Xiao, Proxy multi-signature scheme: a new type of proxy signature scheme, Electronics Letters 36 (6) (2000) 527–528.
[22] K. Zhang, Threshold proxy signature schemes, 1997 Information Security Workshop, 1997, pp. 191–197.