International Journal of Innovative
Computing, Information and Control
Volume 7, Number 2, Feb. 2011
ICIC International ©2011 ISSN 1349-4198
pp. ?–?

GENERALIZATION OF PROXY SIGNATURE BASED ON FACTORIZATION

Cheng-Chi Lee

Department of Photonics and Communication Engineering
Asia University
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.
cclee@asia.edu.tw

Tzu-Chun Lin

Department of Applied Mathematics
Feng Chia University
100 Wenhwa Rd, Seatwen, Taichung, Taiwan 407, R.O.C.

Shiang-Feng Tzeng

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

Min-Shiang Hwang

Department of Management Information Systems
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
Correspondence e-mail: mshwang@nchu.edu.tw

Abstract. *A rich set of proxy signature schemes have been widely researched and discussed so far. However, they have been mainly focusing on dealing with one or two separate proxy situations each. In this article, the authors proposed the generalization of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on the factorization of the square root modulo of a composite number. This scheme 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 of proxy signers. Any verifier can verify the proxy signatures on the messages with the knowledge of the identities of the actual original signers and the actual proxy signers. Furthermore, all possible attacks that have been analyzed so far have failed to break the proposed scheme.*

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

1. **Introduction.** Digital signatures [3, 19, 37] are widely used to replace hand-written signatures in the digital world. However, simple digital signature schemes are not enough to satisfy today's practical conditions. For example, suppose a chairman in a department needs to go on a business trip. She/he has to find a proxy person to deal with her/his work at the office. Traditional digital signature schemes [1, 21, 33, 48] do not meet this requirement. To remedy this weakness, the proxy function has been added to digital signature schemes, and this new type of digital signature is called proxy signature [4, 28, 40, 44].

In 1996, Mambo et al. [30, 31] proposed the first proxy signature schemes. Their schemes allow 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.

As discussed by the authors [20, 30, 31], proxy signature schemes have four delegation types: full delegation, partial delegation, delegation by warrant, and partial delegation with warrant. With full delegation, a proxy signer receives the same private key as the original signer has. In this case, a valid proxy signature by a proxy signer is indistinguishable from a signature generated by the original signer.

With partial delegation, the original signer uses her/his own private key to create a proxy signature key and gives this proxy signature key to the proxy signer. It is computationally infeasible for the proxy signer to obtain the original signer's private key from the given proxy signature key. The proxy signer can use the proxy signature key to sign messages on behalf of the original signer.

With delegation by warrant, a proxy signer receives a proxy warrant. The proxy warrant is used to certify that the proxy signer is delegated by the original signer. The proxy warrant records the identities of the original signer and the proxy signer, the valid proxy period, etc.

Finally, the last delegation kind combines the benefits of both partial delegation and delegation by warrant. The advantages of partial delegation with warrant are fast processing speed and the message qualification. Furthermore, neither delegation by warrant nor partial delegation with warrant needs an additional proxy revocation protocol. Among these four delegation kinds, partial delegation with warrant seems to be the best choice. In this article, we focus on the proxy signature for partial delegation with warrant.

In general, a proxy signature for partial delegation should satisfy the following necessary conditions [30, 31]: strong unforgeability, verifiability, proxy signer's deviation, distinguishable property, strong identifiable property, secret-keys' dependence, and strong undeniable property.

So far, quite a number of proxy signature schemes have been widely discussed [5, 10, 12, 20, 23, 36, 41, 45, 46, 47]. The proxy signature schemes where the original signer can delegate only one proxy signer cannot be used for group proxy situations. As a result, threshold proxy signature schemes ($(1-t/n)$ proxy signature scheme), proxy multi-signature schemes ($(n-1)$ proxy signature scheme), and multi-proxy multi-signature schemes ((n_1-n_2) proxy signature scheme) have also been proposed.

$(1-t/n)$ proxy signature schemes have received much attention [6, 10, 11, 12, 14, 20, 23, 35, **?**]. In such a design, the original signer is able to delegate her/his signing capability to n signers of the proxy group. Any t or more proxy signers in the proxy group can cooperatively generate the proxy signature.

$(n-1)$ proxy signature schemes can be found in [2, 9, 24, 25, 38, 42, 47]. In these schemes, a proxy signer is allowed to generate a proxy signature on behalf of two or more original signers. However, these schemes have a common disadvantage: the size of the proxy signature on the proxy's warrant is proportional to the size of the original signer group.

Recently, a new type of proxy signature, called the (n_1-n_2) proxy signature, was proposed [7, 8, 15, 29, 43]. In this scheme, the original group can delegate a proxy group under the agreement of all the signers in both the original group and proxy group.

In general, the $(n-1)$ proxy signature scheme is a special case of the $(t_1/n_1-t_2/n_2)$ proxy signature scheme where n out of n original signers in the original group and the proxy group has only one single proxy signer. On the other hand, the $(1-t/n)$ 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 and the proxy group consist of only a single original signer and a single proxy singer, it is also a special case of the $(t_1/n_1-t_2/n_2)$ proxy signature scheme. The concept of the $(t_1/n_1-t_2/n_2)$ proxy signature scheme was first introduced by Li et al. [26]. It is an important technique to generalize the proxy signature schemes. Li et al. proposed a generalization of proxy signature based on discrete logarithms that can be applied to every proxy situation. However, Li et al.'s scheme has a security weakness that an adversary can forge illegal proxy signatures being likely generated by the proxy group on behalf of an adversary. To overcome this weakness, Hwang and Chan [17] proposed an improvement on Li et al.'s generation scheme. Based on elliptic curves, Hwang et al. proposed another generalization of proxy signature scheme [16]. However, [18] and [39], respectively, showed that the scheme has security flaws and then proposed an improvement on Hwang et al.'s generation scheme.

Many asymmetric cryptosystems have been proposed, however, only three types of systems should be considered both secure and efficient. These systems, classified according to the mathematical problem on which they are based, are:

- Factorization problem (such as the RSA cryptosystem)

- Discrete Logarithm problem (such as the ElGamal cryptosystem)
- Elliptic Curves Cryptosystem (ECC)

Up to now, two generalization of proxy signature schemes based on discrete logarithms and elliptic curves are proposed by Li et al. and Hwang et al., respectively. In this article, we shall propose another generalization of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme with known signers based on a new type called factorization. The proposed scheme is suitable for all the proxy situations mentioned above.

This article is organized as follows. In Section 2, we shall first briefly review Lee et al.'s scheme (the Lee-Hwang-Li scheme). Then, in Section 3, our new secure $(t_1/n_1 - t_2/n_2)$ proxy signature scheme and security analysis will be proposed and presented. After that, we propose some special cases of the $(t_1/n_1 - t_2/n_2)$ proxy signature in Section 4. Finally, some concluding remarks will be in the last section.

2. **Review of the Lee-Hwang-Li Digital Signature Scheme.**

Lee et al. [22] proposed a digital signature scheme based on the Ohto-Okamoto scheme [32]. Initially, each user randomly generates two large primes, p' and q', and computes p, q, N, and $\lambda(N)$ as follows: $p = 2p' + 1$, $q = 2q' + 1$, $N = p \times q$, and $\lambda(N) = 2p'q'$. Next, each user selects a secret random number s_i between 1 and $\lambda(N)$. Then she/he computes a private key $x_i = \alpha^{s_i} \bmod N$ and the corresponding public key y_i which is certified by the certificate authority (CA) as $y_i = \alpha^{-s_i L} = x_i^{-L} \bmod N$, where L is a random number ($\approx 10^{50}$) with $GCD(L, \lambda(N)) = 1$, α is an element which is primitive in both $GF(p)$ and $GF(q)$, and $h(\cdot)$ is a one-way hash function. The parameters N, L, and $h(\cdot)$ are published, and p, q, p', q', α, and $\lambda(N)$ are kept secret.

Assume the message M needs to be signed by the signer. The signer randomly selects a random number k between 1 and $(N - 1)$. Then, she/he computes

$$
\begin{aligned}
r &= k^L \bmod N, \\
e &= h(M, r) \bmod N, \\
z &= k x_i^e \bmod N.
\end{aligned}
$$

Then, (e, z) is a signature of M. The signer sends (e, z) with M to the verifier.

On receiving the digital signature (e, z) of M, the verifier can compute r' as $r' = z^L y_i^e \bmod N$. Then the verifier can check the equation, $e \stackrel{?}{=} h(M, r')$. If the equation holds, (e, z) is a valid signature of M.

3. **The Proposed Scheme.**

In this section, we propose and analyze a new $(t_1/n_1 - t_2/n_2)$ proxy signature scheme with known signers. This scheme is based on the Lee-Hwang-Li's digital signature scheme as described previously.

According to the proxy warrant, a subset of original signers allows a designated proxy group to sign on behalf of the original group. A message M has to be signed by a subset of proxy signers who can represent the proxy group. Then, the proxy signature is sent to a verifier to verify the validity of it. In other words, some threshold values will be given to indicate the number of persons to represent the group to delegate the signing capability or to sign the proxy signature.

Assume that there is a share distribution center (SDC) which is responsible for distributing shares for the system. Initially, the system parameters (N, L, $h(\cdot)$, p, q, p', q', α, $\lambda(N)$) are the same as those of the Lee-Hwang-Li digital signature scheme back in Section 2. Now, we will set some system parameters as:

- M_w*:* a warrant which records the identities of the original signers in the original group and the proxy signers in the proxy group, 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.

Next, SDC computes the original signer's private key x_{O_i} and public key y_{O_i} as follows:

$$
\begin{aligned}
x_{O_i} &= \alpha^{s_{O_i}} \bmod N, \\
y_{O_i} &= x_{O_i}^L \bmod N,
\end{aligned}
\tag{1}
$$

where s_{O_i} is a random integer such that $GCD(s_{O_i}, \lambda(N)) = 1$. Similarly, each proxy signer U_{P_i} also owns a private key $x_{P_i} = \alpha^{s_{P_i}} \bmod N$, where s_{P_i} is a random integer such that $GCD(s_{P_i}, \lambda(N)) = 1$, and a public key:

$$y_{P_i} = x_{P_i}^L \bmod N, \tag{2}$$

which is also certified by that CA. Finally, SDC distributes private keys to all the original signers and proxy singers. The notations G_O and G_P are defined as the original signer group and the proxy signer group, respectively. 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.

Based on the definition of the $(t_1/n_1 - t_2/n_2)$ proxy signature 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 signing capability. 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. The procedure of the proposed scheme contains three phases: the proxy share generation phase, the proxy signature generation phase, and the proxy signature verification phase.

3.1. **Proxy Share Generation Phase.**

Assume that 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 with $t_1 \le t'_1 \le n_1$, and D_O as a group executes the following steps to delegate the signing capability to G_P.

1. Choose a random number k_{O_i} with $0 < k_{O_i} < N$, and broadcast r_{O_i} to other $t'_1 - 1$ original signers in D_O and the designated clerk.

$$r_{O_i} = k_{O_i}^L \bmod N \tag{3}$$

2. For each received r_{O_j} $(j = 1, 2, \cdots, t'_1; j \ne i)$, each $U_{O_i} \in D_O$ computes K and σ_{O_i} as follows:

$$K = \prod_{i=1}^{t'_1} r_{O_i} \bmod N, \tag{4}$$

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

3. Send σ_{O_i} to the designated clerk via a public channel.
4. After receiving σ_{O_i}, the designated clerk checks whether the following equation holds:

$$\sigma_{O_i}^L = r_{O_i}^K y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod N. \tag{6}$$

If it does, the designated clerk computes σ_O as their proxy share.

$$\sigma_O = \prod_{i=1}^{t'_1} \sigma_{O_i} \bmod N. \tag{7}$$

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:

$$\sigma_O^L = K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod N. \tag{8}$$

After each U_{P_j} confirms the validity of $(M_w, K, \sigma_O, AOSID)$, she/he computes her/his proxy signature key σ_{P_j} as

$$\sigma_{P_j} = \sigma_O x_{P_j}^{y_{P_j}} \bmod N. \tag{9}$$

3.2. **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 with $t_2 \le t'_2 \le n_2$, and D_P as a group executes the following steps to generate a proxy signature.

1. Choose a random number k_{P_j} with $0 < k_{P_j} < N$ and broadcast r_{P_j} to other $t'_2 - 1$ proxy signers in D_P and the designated clerk.
$$r_{P_j} = k_{P_j}^L \bmod N.$$
2. For each received r_{P_k} $(k = 1, 2, \cdots, t'_2; k \neq j)$, each $U_{P_j} \in D_P$ computes R and s_{P_j} as follows:
$$R = \prod_{j=1}^{t'_2} r_{P_j} \bmod N, \tag{10}$$
$$s_{P_j} = k_{P_j}^R \sigma_{P_j}^{h(M,R,APSID)} \bmod N. \tag{11}$$
Here, s_{P_j} is an individual proxy signature which is sent to the designated clerk.
3. After receiving s_{P_j}, the designated clerk checks whether the following equation holds:
$$s_{P_j}^L = r_{P_j}^R (K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N. \tag{12}$$
If it does, the designated clerk computes
$$S = \prod_{j=1}^{t'_2} s_{P_j} \bmod N. \tag{13}$$

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

3.3. **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 can check the validity of the proxy signature on the message M through the following equation:
$$S^L = R^R((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t'_2} \prod_{j=1}^{t'_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N. \tag{14}$$

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

In the following paragraphs, we shall prove that the proposed scheme can work correctly.

Theorem 3.1. *In the proxy share generation phase, the designated clerk can verify the validity of σ_{O_i} sent from U_{O_i} by Equation (6).*

Proof. From Equation (5), we have $\sigma_{O_i} = k_{O_i}^K x_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \bmod N$. By raising both sides of the above equation to exponents with L, we have
$$\sigma_{O_i}^L = k_{O_i}^{LK} x_{O_i}^{L y_{O_i} h(M_w,K,AOSID)} \bmod N.$$
According to Equations (1) and (3), the above equation can be rewritten as
$$\sigma_{O_i}^L = r_{O_i}^K y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \bmod N.$$
□

Theorem 3.2. *If the proxy share is constructed correctly, it will pass the verification of Equation (8).*

Proof. From Equations (5) and (7), we have

$$\begin{aligned}\sigma_O &= \prod_{i=1}^{t'_1} \sigma_{O_i}, \\ &= \prod_{i=1}^{t'_1} k_{O_i}^K x_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \bmod N.\end{aligned}$$

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

$$\begin{aligned}\sigma_O^L &= \prod_{i=1}^{t'_1} k_{O_i}^{LK} x_{O_i}^{Ly_{O_i} h(M_w,K,AOSID)}, \\ &= \prod_{i=1}^{t'_1} r_{O_i}^K y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \bmod N.\end{aligned}$$

In accordance with Equation (4), the above equation can be rewritten as

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

□

Theorem 3.3. *In the proxy signature generation phase, the designated clerk can verify any individual proxy signature s_{P_i} sent from U_{P_i} by Equation (12).*

Proof. Taking in Equation (9), we can rewrite Equation (11) as

$$s_{P_j} = k_{P_j}^R (\sigma_O x_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.$$

Raising both sides of the above equation to exponents with L, we have

$$\begin{aligned}s_{P_j}^L &= k_{P_j}^{LR} (\sigma_O x_{P_j}^{y_{P_j}})^{Lh(M,R,APSID)}, \\ &= r_{P_j}^R (\sigma_O^L y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.\end{aligned}$$

According to Equation (8), the above equation can be rewritten as

$$s_{P_j}^L = r_{P_j}^R (K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.$$

□

Theorem 3.4. *If the proxy signature is constructed correctly, it will pass the verification of Equation (14).*

Proof. In Equations (9), (11) and (13), we have

$$\begin{aligned}S &= \prod_{j=1}^{t'_2} s_{P_j}, \\ &= \prod_{j=1}^{t'_2} k_{P_j}^R (\sigma_O x_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.\end{aligned}$$

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

$$S^L = \prod_{j=1}^{t'_2} r_{P_j}^R (\sigma_O^L y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.$$

According to Equations (8) and (10), the above equation can be written as

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

$\square$

3.4. **Security Analysis and Discussions.**

In this subsection, we analyze the security of the proposed scheme and prove that the proposed scheme can satisfy the conditions in [30, 31].

The security of the proposed scheme is based on the well-known difficulty of computing one-way hash function [13] and the factorization [34] of the square root modulo of a composite number N. In the following, we consider some possible attacks and see how the proposed scheme works against them. We shall prove that the proposed scheme can successfully withstand these attacks, maintaining the security provided by the one-way hash function and the factorization problem of N.

In the proposed scheme, all of the actual original signers' private keys x_{O_i} are used in the proxy share generation phase to create a proxy share σ_O. Therefore, for the proxy group, it is necessary to verify the proxy share σ_O by checking Equation (8), and the verifier also has to check the proxy signature verification equation, namely Equation (14), by using all of the actual original signers' public keys. All of the actual original signers' public keys have been certified by CA. Without knowing the original signers' private keys, an adversary is unable to create the proxy share σ_O in the proposed scheme. Assume that an adversary wants to obtain the original signer's private key x_{O_i} in Equation (1). It means the adversary has to face the difficulty of solving the factorization of N. According to the warrant M_w, the original group cannot deny delegating their signing capability to a proxy group because of the proof of the proxy share.

Similarly, all of the actual proxy signers' private keys x_{P_i} are used to generate the proxy signature in the proxy signature generation phase. Thus, it is necessary for any verifier to verify the proxy signature verification equation, which is Equation (14), by using all of the actual proxy signers' public keys. These actual proxy signers' public keys have also been certified by CA. If an adversary wants to get some proxy signer's private key x_{P_i} from Equation (2), then she/he has to solve the factorization of N. Therefore, the proxy group cannot deny generating the proxy signature on behalf of the proxy group and the original group.

To fight against the equation attack, we have to consider the security of the proxy signature verification equation:

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

In this case, an adversary may try to forge a valid proxy signature to pass the proxy signature verification equation. Suppose

$$\begin{aligned} V_O &= (K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)})^{t'_2} \bmod N, \qquad (15) \\ V_P &= \prod_{j=1}^{t'_2} y_{P_j}^{y_{P_j}} \bmod N. \end{aligned}$$

We can rewrite the proxy signature verification equation as

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

The value V_O depends on the parameters M_w, K and $AOSID$, while 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' based on the one-way hash function and the factorization of N. Again, under given M', $APSID'$, V'_P, R' and S', V'_O is to be computed so that the equation holds. However, it is difficult to find M'_w, K' and $AOSID'$ such that Equation (15) holds. It also requires solving the one-way hash function and the factorization of N. Therefore, the proxy signature verification equation is secure against the equation attack.

Consider the insider forgery attack in [27, 38]. Without losing of generality, the malicious original signer U_{O_k} ($U_{O_k} \in G_O$) without any private keys of the other original signers can forge a valid proxy share for an arbitrary warrant. To forge the proxy share, assume U_{O_k} waits until she/he receives any $t_1 - 1$ original signers' public keys y_{O_i}. She/he substitutes her/his public key y_{O_k}. U_{O_k} selects an integer x'_{O_k} as her/his private key. Then, U_{O_k} has to make her/his public key y'_{O_k} satisfy the following equation:

$$y'^{y'_{O_k}}_{O_k} = x'^{L}_{O_k} (\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{-1} \bmod N. \tag{16}$$

Next, U_{O_k} selects a random integer k_{O_k} and computes K and σ_O as follows:

$$\begin{aligned} K &= k_{O_k}^{L} \bmod N, \\ \sigma_O &= k_{O_k}^{K} x'^{h(M_w,K,AOSID)}_{O_k} \bmod N. \end{aligned}$$

Thus, σ_O is a valid proxy share. It is because the forged proxy share can pass Equation (8) as follows:

$$\begin{aligned} \sigma_O^L &= K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)}, \\ &= K^K (x'^{L}_{O_k} (\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{-1} \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}})^{h(M_w,K,AOSID)}, \\ &= (k_{O_k}^{K} x'^{h(M_w,K,AOSID)}_{O_k})^L \bmod N. \end{aligned}$$

However, U_{O_k} cannot succeed in generating the proxy share σ_O. From Equation (16), suppose the values of x'_{O_k} and $\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i}} \bmod N$ are determined, it is extremely difficult to find a y'_{O_k} satisfying the equation. If U_{O_k} determines the integer y'_{O_k} first, she/he has to solve the factorization of N to find the value of x'_{O_k}. Thus, the insider forgery attack launched by the malicious original signer in the original group cannot succeed in forging the proxy share for the arbitrary warrant.

Similarly, G_P might also try the insider forgery attack. Without loss of generality, suppose that the malicious proxy signer U_{P_k} ($U_{P_k} \in G_P$) 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:

$$y'^{y'_{P_k}}_{P_k} = x'^{L}_{P_k} \times (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{-1} \bmod N. \tag{17}$$

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

$$\begin{aligned} R &= r_{P_k}^{L} \bmod N, \\ S &= r_{P_k}^{R} (\sigma_O^{t_2} x'_{P_k})^{h(M,R,APSID)} \bmod N. \end{aligned}$$

Thus, (M_w, K, $AOSID$, M, R, S, $APSID$) is a valid proxy signature. It is because the forged proxy signature can satisfy Equation (14) as follows:

$$\begin{aligned} S^L &= R^R((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\ &= R^R((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} x'^{L}_{P_k} (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{-1} \prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\ &= R^R((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} x'^{L}_{P_k})^{h(M,R,APSID)}, \\ &= (r_{P_k}^{R} (\sigma_O^{t_2} x'_{P_k})^{h(M,R,APSID)})^L \bmod N. \end{aligned}$$

However, U_{P_k} cannot create a valid proxy signature. From Equation (17), if the values of x'_{P_k} and $\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}}$ mod N are determined, it is an extremely difficult problem to find a y'_{P_k} satisfying the equation. If U_{P_k} determines the integer y'_{P_k} first, she/he has to solve the factorization of N to find the value of x'_{P_k}. Thus, the insider forgery attack launched by a malicious proxy signer cannot work.

As for the direct forgery attack in [38], the security analysis is the same as that of the insider forgery attack for the proposed scheme. A malicious original signer can impersonate any $t_1 - 1$ or more original signers in the original group. The malicious original signer can forge the proxy share without relying on the agreement of the other original signers. Suppose U_{O_k} waits until she/he receives any $t_1 - 1$ original signers' public keys. Then, U_{O_k} selects a random number k_{O_k} and computes K and σ_O as follows:

$$\begin{aligned} K^K &= k_{O_k}^L \Big(\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}\Big)^{-1} \bmod N, \\ \sigma_O &= k_{O_k} x_{O_k}^{y_{O_k} h(M_w, K, AOSID)} \bmod N. \end{aligned} \tag{18}$$

Thus, σ_O is a valid proxy share because the forged proxy share can pass Equation (8) as follows:

$$\begin{aligned} \sigma_O^L &= K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}, \\ &= k_{O_k}^L \Big(\prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}\Big)^{-1} y_{O_k}^{y_{O_k} h(M_w, K, AOSID)} \prod_{i=1}^{t_1-1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}, \\ &= k_{O_k}^L y_{O_k}^{y_{O_k} h(M_w, K, AOSID)}, \\ &= (k_{O_k} x_{O_k}^{y_{O_k} h(M_w, K, AOSID)})^L \bmod N. \end{aligned}$$

However, U_{O_k} cannot succeed in generating the proxy share σ_O. From Equation (18), it is extremely difficult to find a K satisfying the equation. If U_{O_k} determines the integer K first, she/he has to solve the factorization of N to find the value of k_{O_k}. Thus, the direct forgery attack launched by the malicious original signer in the original group cannot succeed.

On the other hand, a malicious proxy signer can also impersonate any $t_2 - 1$ or more proxy signers in the proxy group. U_{P_k} can also forge the proxy signature without the agreement of the other proxy signers. Suppose that U_{P_k} waits until she/he receives any $t_2 - 1$ proxy signers' public keys. Then, U_{P_k} selects a random number r_{P_k} and computes R and S as follows:

$$\begin{aligned} R^R &= r_{P_k}^L \Big(\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j} h(M, R, APSID)}\Big)^{-1} \bmod N, \\ S &= r_{P_k} (\sigma_O^{t_2} x_{P_k}^{y_{P_k}})^{h(M, R, APSID)} \bmod N. \end{aligned} \tag{19}$$

Thus, (M_w, K, $AOSID$, M, R, S, $APSID$) is a valid proxy signature because the forged proxy signature can satisfy Equation (14) as follows:

$$\begin{aligned}
S^L &= R^R((K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} \prod_{j=1}^{t_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= r_{P_k}^L (\prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j} h(M,R,APSID)})^{-1}((K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} \times \\
&\quad y_{P_k}^{y_{P_k}} \prod_{j=1}^{t_2-1} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)}, \\
&= r_{P_k}^L ((K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2} y_{P_k}^{y_{P_k}})^{h(M,R,APSID)}, \\
&= (r_{P_k}(\sigma_O^{t_2} x_{P_k}^{y_{P_k}})^{h(M,R,APSID)})^L \bmod N.
\end{aligned}$$

However, U_{P_k} still cannot create a valid proxy signature. From Equation (19), it is extremely difficult to find a R satisfying the equation. If U_{P_k} determines the integer R first, she/he has to solve the factorization of N to find the value of r_{P_k}. Thus, the direct forgery attack launched by the malicious proxy signer cannot work.

The proposed scheme satisfies the proxy signature scheme requirements in [30, 31]. The analysis is given below.

- *Strong unforgeability:* A designated proxy group can generate a valid proxy signature for the original group. According to the above security analysis, any original signer in the original group and any third party cannot generate a valid proxy share and a valid proxy signature. So, the proposed scheme satisfies the strong unforgeable requirement. Furthermore, the proposed scheme provides proxy-protection property.
- *Verifiability:* By the proxy signature, any verifier can be convinced of the original group's agreement on the signed message. The verification of the proxy signature can be used to guarantee that the original group actually authorizes the proxy group. Hence, the proposed scheme satisfies the verifiability requirement.
- *Proxy signer's deviation:* The proxy signers cannot generate a valid proxy signature without being detected as a proxy signature. The proxy signers cannot generate the valid signatures of the original signers. In the other words, the proposed scheme provides the proxy signer's deviation.
- *Distinguishability:* The valid proxy signature and the self-signing signature are applied to different congruences for verification. Here, a self-signing signature refers to an ordinary signature generated by the original signers. Therefore, these verification congruences can distinguish the proxy signature from the self-signing signature. The proposed scheme therefore achieves distinguishable requirement.
- *Strong identifiability:* From the proxy signature, anyone can determine the identity of the corresponding proxy signers. Furthermore, from the $APSID$, anyone gets to know with ease who actually signed the message on behalf of the proxy group. Besides, from $AOSID$, we easily know who actually delegated the signing power on behalf of the original group. Hence, the proposed scheme satisfies the Strong identifiable requirement.
- *Secret-keys' dependence:* The proxy signature is dependent on the proxy signers' private keys and the proxy share σ_O. The proposed scheme provides the secret-keys' dependence through the proxy share.
- *Strong undeniability:* The actual proxy signers generate a valid proxy signature for the original signer group and the proxy signer group. The actual proxy signers cannot repudiate the creation of the signature. Therefore, the proposed scheme provides the strong undeniable property.

4. **Special Cases of the Proposed Scheme.**

In this section, we propose some special cases in the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme. Basically, these special cases are realized in the framework of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme. We propose three kinds of proxy signature schemes: the $(t/n-1)$ proxy signature scheme, the $(1-t/n)$ proxy signature scheme, and the $(1-1)$ proxy signature scheme.

4.1. **The $(t/n - 1)$ Proxy Signature Scheme.**

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

The system parameters are the same as those in Section 3. 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 3. 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 (8) holds. If it does, U_P confirms the validity of $(M_w, K, \sigma_O, AOSID)$. She/he computes her/his proxy signature key $\sigma_P = \sigma_O x_P^{y_P} \bmod N$.

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 chooses a number r and computes

$$\begin{aligned} R &= r^L \bmod N, \\ S &= r^R \sigma_P^{h(M,R)} \bmod N. \end{aligned}$$

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 of the original group through the following equation:

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

with $t \le t' \le n$. If the above equation holds, the verifier believes in the validity of the proxy signature and know the identities of the actual original signers from $AOSID$.

4.2. **The $(1 - t/n)$ Proxy Signature Scheme.**

In the $(1-t/n)$ proxy signature scheme, any t or more proxy signers from a designated proxy group of n 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 in Section 3. 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 chooses a random number k and computes

$$\begin{aligned} K &= k^L \bmod N, \\ \sigma_O &= k^K x_O^{y_O h(M_w, K)} \bmod N. \end{aligned}$$

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

$$\sigma_O^L = K^K y_O^{y_O h(M_w, K)} \bmod N. \tag{20}$$

If it does, each U_{P_j} uses σ_O to compute her/his proxy signature key σ_{P_j} in Equation (9).

In the proxy signature generation phase, any t or more proxy signers, as a group, signs a message M cooperatively on behalf of G_P. The procedure is similar to that in Section 3. Each U_{P_j} computes R and s_{P_j} in Equations (10) and (11) and sends s_{P_j} to the designated clerk. The clerk can verify the validity of s_{P_j} according to the following equation:

$$s_{P_j}^L = r_{P_j}^R (K^K y_O^{y_O h(M_w, K)} y_{P_j}^{y_{P_j}})^{h(M, R, APSID)} \bmod N.$$

If the equation holds, then the clerk derives S from Equation (13), 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 of the proxy group through the following equation:

$$S^L = R^R((K^K y_O^{y_O h(M_w,K)})^{t'} \prod_{j=1}^{t'} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N,$$

with $t \le t' \le n$. If the above equation holds, the verifier can make sure of the validity of the proxy signature and identify the actual proxy signers from $APSID$.

4.3. 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 in Section 3. 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 4.2. 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 (20), U_P performs the same steps as those of the proxy signature generation phase in Section 4.1. Then, the proxy signature of message M is (M_w, K, M, R, S). Finally, the verifier checks the validity of the proxy signature in the following equation:

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

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

The security analysis and necessary conditions of the $(t/n-1)$, $(1-t/n)$ and $(1-1)$ proxy signature schemes presented above are the same as those in Section 3. Those attacks in the above schemes will not work here because it is difficult to obtain the proxy share or the proxy signature. The proposed schemes also satisfy those necessary conditions.

5. Conclusions.

In this paper, we have proposed a new generalization of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on factorization to overcome security weaknesses. Based on the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme, three special cases, namely the $(t/n-1)$ proxy signature scheme, the $(1-t/n)$ proxy signature scheme and the $(1-1)$ proxy signature scheme, have been explored. For avoiding the abuse of signing capability, the proposed scheme provides the ability to identify the actual original signers and the actual proxy signers. These actual original signers cannot deny delegating the warrant. Furthermore, these actual proxy signers cannot deny signing the proxy signatures. 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. In practice, the proposed proxy signature scheme can be applied to a company or an organization. If a manager wants to go on a business trip, he/she 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. Until to now, generalization of proxy signature schemes based on discrete logarithms, elliptic curves, and factorization are not efficient. In future, we can try to improve their efficiency.

Acknowledgment. The authors would like to thank the anonymous referee for their valuable discussions and comments. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC NSC96-2219-E-009-013 and NSC98-2221-E-468-002.

REFERENCES

[1] F. Bao, C. C. Lee, and M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple RSA digital signatures," *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 1195–1200, 2006.

[2] T. S. Chen, Y. F. Chung, and G. S. Huang, "Efficient proxy multisignature schemes based on the elliptic curve cryptosystem," *Computer & Security*, vol. 22, no. 6, pp. 527–534, 2003.

[3] Yu-Fang Chung and Kuo-Hsuan Huang, "Chameleon signature with conditional open verification," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 9, pp. 2829–2836, 2009.

[4] Manik Lal Das, Ashutosh Saxena, Deepak B. Phatak, "Proxy signature scheme with effective revocation using bilinear pairings," *International Journal of Network Security*, vol. 4, no. 3, pp. 312–317, 2007.

[5] Manik Lal Das, Ashutosh Saxena, and Deepak B Phatak, "Algorithms and approaches of proxy signature: A survey," *International Journal of Network Security*, vol. 9, no. 3, pp. 264-284, 2009.

[6] Chao-Sheng Feng, Zhi-Guang Qin, and Ding Yuan, "Improvement of a threshold proxy signature scheme," in *International Conference on Machine Learning and Cybernetics*, pp. 3168–3172, Aug. 2007.

[7] L. Guo and G. Wang, "Insider attacks on multi-proxy multi-signature schemes," *Computers and Electrical Engineering*, vol. 33, no. 2, pp. 88–93, 2007.

[8] C. L. Hsu, K. Y. Tsai, and P. L. Tsai, "Cryptanalysis and improvement of nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Information Sciences*, vol. 177, no. 2, pp. 543–549, 2007.

[9] C. L. Hsu, T. S. Wu, and W. H. He, "New proxy multi-signature scheme," *Applied Mathematics and Computation*, vol. 162, no. 3, pp. 1201–1206, 2005.

[10] 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.

[11] H. F. Huang and C. C. Chang, "A novel efficient (t,n) threshold proxy signature scheme," *Information Sciences*, vol. 176, no. 10, pp. 1338–1349, 2006.

[12] 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.

[13] Min-Shiang Hwang, Chin-Chen Chang, and Kuo-Feng Hwang, "A watermarking technique based on one-way hash functions," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 2, pp. 286–294, 1999.

[14] 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*, vol. 15, no. 6, pp. 1552–1560, 2003.

[15] Min-Shiang Hwang, Shiang-Feng Tzeng, and Shu-Fen Chiou, "A non-repudiable multi-proxy multi-signature scheme," *ICIC Express Letters*, vol. 3, no. 3, pp. 259–264, 2009.

[16] M. S. Hwang, S. F. Tzeng, and C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards and Interfaces*, vol. 26, pp. 73–84, March 2004.

[17] S. J. Hwang and C. C. Chan, "Improvement on Li et al.'s generalization of proxy signature schemes," *Computers and Security*, vol. 23, pp. 615–619, Oct. 2004.

[18] S. J. Hwang and C. W. Huang, "Improvement on Hwang et al.'s generalization of proxy signature schemes based on elliptic curves," *Applied Mathematics and Computation*, vol. 170, pp. 941–947, Nov. 2005.

[19] R. S. Katti and R. G. Kavasseri, "Nonce Generation For The Digital Signature Standard," *International Journal of Network Security*, vol. 11, no. 1, pp. 23-32, 2010.

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

[21] C. C. Lee, M. S. Hwang, and W. P. Yang, "Untraceable blind signature schemes based on the discrete logarithm problem," *Fundamenta Informaticae*, vol. 55, pp. 307–320, May-June 2003.

[22] Narn-Yih Lee, Tzonelih Hwang, and Chuan-Ming Li, "*(t, n)* threshold untraceable signatures," *Journal of Information Science and Engineering*, vol. 16, no. 6, pp. 835–845, 2000.

[23] 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.

[24] F. G. Li, S. J. Zhou, and R. Sun, "Cryptanalysis of an identity based proxy multi-signature scheme," *IEICE Tran. on Fundamentals of Electronics Communications and Computer Sciences*, vol. E91A, pp. 1820–1823, Jul. 2008.

[25] J. G. Li, Z. F. Cao, and Y. C. Zhang, "Nonrepudiable proxy multi-signature scheme," *Journal of Computer Science and Technology*, vol. 18, pp. 399–402, May 2003.

[26] L. H. Li, S. F. Tzeng, and M. S. Hwang, "Generalization of proxy signature-based on discrete logarithms," *Computers and Security*, vol. 22, pp. 245–255, April 2003.

[27] 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, pp. 314–315, 2000.

[28] Eric Jui-Lin Lu and Cheng-Jian Huang, "A time-stamping proxy signature scheme using time-stamping service," *International Journal of Network Security*, vol. 2, no. 1, pp. 43–51, 2006.

[29] Y. D. Lyuu and M. L. Wu, "Cryptanalysis of and improvement on the Hwang-Chen multi-proxy multi-signature schemes," *Applied Mathematics and Computation*, vol. 167, pp. 729–739, Aug. 2005.

[30] 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.

[31] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," *Proc. Third ACM Conf. on Computer and Communications Security*, pp. 48–57, 1996.

[32] Kazuo Ohta and Tatsuaki Oktamoto, "A modification of the Fiat-Shamir scheme," in *Advances in Cryptology, CRYPTO'88*, pp. 232–243, Lecture Notes in Computer Science, 1988.

[33] R. Rajaram Ramasamy and M. Amutha Prabakar, "Digital Signature Scheme with Message Recovery Using Knapsack-based ECC," *International Journal of Network Security*, vol. 12, no. 1, pp. 12-17, 2011.

[34] 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.

[35] J. Shao, Z. Cao, and R. Lu, "Improvement of Yang et al.s threshold proxy signature scheme," *The Journal of Systems & Software*, vol. 80, no. 2, pp. 172–177, 2007.

[36] Z. Shao, "Proxy signature schemes based on factoring," *Information Processing Letters*, vol. 85, no. 3, pp. 137–143, 2003.

[37] Wen-Gong Shieh and Mei-Tzu Wang, "An improvement to Kim-Chung's authentication scheme," *ICIC Express Letters*, vol. 3, no. 4, pp. 1215–1220, 2009.

[38] H. M. Sun, "On proxy (multi-) signature schemes," in *2000 International Computer Symposium*, pp. 65–72, Chiayi, Taiwan, Dec. 2000.

[39] Zuowen Tan, "Improvement on a generalized scheme of proxy signature based on elliptic curves," in *International Conference on Computational Intelligence and Security (CIS 2007)*, pp. 677–681, 2007.

[40] Zuo-Wen Tan, "Improvement on Nominative Proxy Signature Schemes," *International Journal of Network Security*, vol. 7, no. 2, pp. 175–180, 2008.

[41] G. K. Verma, "A proxy blind signature scheme over braid groups," *International Journal of Network Security*, vol. 9, no. 3, pp. 214-217, 2009.

[42] Q. Wang and Z. F. Cao, "Identity based proxy multi-signature," *The Journal of Systems and Software*, vol. 80, no. 7, pp. 1023–1029, 2007.

[43] Z. C. Wang, H. F. Qlan, and Z. B. Li, "Hybrid proxy multisignature: a new type multi-party signature," *Information Sciences*, vol. 177, no. 24, pp. 5638–5650, 2007.

[44] Qi Xie and Xiu-Yuan Yu, "Cryptanalysis of two nonrepudiable threshold proxy signature schemes," *International Journal of Network Security*, vol. 3, no. 1, pp. 18–22, 2006.

[45] Hu Xiong, Zhiguang Qin, and Fagen Li, "A certificateless proxy ring signature scheme with provable security," *International Journal of Network Security*, vol. 12, no. 1, pp. 113-127, 2011.

[46] Q. S. Xue and Z. F. Cao, "Factoring based proxy signature schemes," *Journal of Computational and Applied Mathematics*, vol. 195, no. 1-2, pp. 229–241, 2006.

[47] 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.

[48] Jen-Ho Yang and Chin-Chen Chang, "An efficient fair electronic payment system based upon non-signature authenticated encryption scheme," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 11, pp. 3861–3874, 2009.