A New Proxy Signature Scheme for a Specified Group of Verifiers *

Min-Shiang Hwang† Cheng-Chi Lee‡ Shiang-Feng Tzeng§

Department of Computer Science and Information Engineering†
Asia University
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

Department of Library and Information Science‡
Fu Jen Catholic University
510 Jhongjheng Rd., Sinjhuang City,
Taipei County 24205, Taiwan, R.O.C.
and Department of Photonics and Communication Engineering
Asia University
No. 500, Lioufeng Road, Wufeng Shiang, Taichung, Taiwan, R.O.C.
Corresponding Email: cclee@mail.fju.edu.tw

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

January 21, 2013

*This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grant NSC 96-2219-E-009-013 and NSC 99-2221-E-030-022.

§Responsible for correspondence: Dr. Cheng-Chi Lee.

A New Proxy Signature Scheme for a Specified Group of Verifiers

Abstract

In this article, we shall propose a $((t_1, n_1), (t_2, n_2))$ proxy signature scheme with (t_3, n_3) shared verification based on the RSA problem. In this scheme, any t_1 original signers can delegate the signing capability to the proxy group. After that, any t_2 proxy signers can sign a message on behalf of the original group for a specified verifier group. Only any t_3 verifiers together can check the validity of the proxy signature from the proxy group. The proposed scheme satisfies all proxy requirements of proxy signatures. Furthermore, the actual original signers and proxy signers can be individually identified in our scheme.

Keywords: Data security, digital signature, proxy signature, threshold proxy signature.

1 Introduction

Digital signature is developed to enable a signer to generate the signature for a message by using her/his private key [2, 13, 32, 34, 37, 40, 54]. To check the validity of the signature, the corresponding public key of the signer should be employed. Generally speaking, a digital signature scheme should provide such important cryptographic function as authentication, integrity and non-repudiation. However, ordinary digital signature schemes [7, 8, 10, 20, 29, 41] are not quite enough to satisfy some practical needs. Let us consider a typical example of the proxy situation in a business organization. Suppose a manager in a computer company needs to go on a business trip. In other words, she/he is not at her/his company and thus is not able to do the routine of signing a number of documents. So, she/he indeed needs a capable and trustworthy secretary to do it instead. Now, the secretary becomes a proxy signer on behalf of the manager. This is a typical case of what happens in our lives every day. Mambo et al. [35, 36] first considered this interesting problem in 1996. It is referred to as the "proxy signature". The "proxy signature" provides a solution to the problem with the delegation of signing capability. The designated proxy signer generates a signature on behalf of the original signer. So far, many proxy signature schemes have been proposed and discussed [11, 14, 15, 18, 22, 23, 25, 26, 27, 28, 45, 47, 48, 51, 59].

However, all of the proxy signature schemes proposed so far allow any outsider to play the role of a verifier. Only one verifier can verify the validity of the proxy signature [5, 33, 44, 55, 57]. In fact, in real-life applications, a signature usually has to go through some specified verifiers. For example, suppose two network companies have a contract between them to sign for commercial collaboration. The contract needs to be signed and verified between the two companies. Assume several directors, called original signers, represent a directorate to delegate their signing capability to some of the designated managers, called proxy signers. Then, several of the designated managers can represent their company to sign this contract with the representative(s) of the other company through a computer network. However, it should never be necessary to check the legitimacy of the signature on occasions when fewer than t_3 of the specified verifiers are present. Only a specified group of verifiers can verify the validity of the contract.

According to the above statement, Tzeng et al. [51] proposed a novel variation of proxy signature scheme called threshold multi-proxy multi-signature scheme with shared verification. Its scheme is based on discrete logarithms problems [10, 19]. In this article, we shall propose a new proxy signature scheme based on RSA problem [6, 21, 41]. To be specific, the authorized message can be delegated, signed and verified by predefined threshold values and under the predefined proxy warrant, respectively.

In the next section, we shall briefly review some related works. In Section 3, we shall propose our new proxy signature scheme with shared verification ((t_1, n_1), (t_2, n_2), (t_3, n_3) proxy signature). In Section 4, the security analysis and performance evaluation of the proposed scheme will be discussed. Finally, the conclusion will be given in the last section.

2 Related Works

So far, five types of proxy delegation have been developed. In 1996, Mambo et al. [35, 36] proposed three types of delegation: the full delegation, the partial delegation and the delegation with warrant. Each of them has its own level of delegation and security assumption. After that, Kim et al. [28] presented two types of delegation: the partial delegation with warrant and threshold delegation. Partial delegation and delegation by warrant are the most secure, and full delegation is the least secure. The advantage of partial delegation is its fast processing speed. In addition, delegation by warrant is appropriate for the restricted period to be signing. Moreover, the warrant can

also be used to prevent a proxy signer from transferring a proxy delegation to another person who is not the designated proxy signer. Partial delegation with warrant combines the benefits of partial delegation and delegation by warrant. Therefore, among these types of delegation, partial delegation with warrant seems to be the best choice. In this article, we shall focus on the proxy signature authorized by using partial delegation with warrant.

In general, a secure proxy signature scheme should satisfy the following security requirements.

- **Strong unforgeability:** A designated proxy signer can generate a valid proxy signature on behalf of the original signer, but the original signer or any third party who is not the designated proxy signer cannot generate a valid proxy signature.
- **Verifiability:** By checking the proxy signature, the verifier can make sure of the original signer's guarantee on the signed message.
- **Proxy signer's deviation:** A proxy signer cannot generate a valid proxy signature not detected as her/his signature. Furthermore, she/he cannot generate a valid signature of the original signer, either.
- **Distinguishability:** Valid proxy signatures are distinguishable from valid self-signing signatures for anyone in polynomial time or size computation. Here, a self-signing signature means a signature generated by the original signer.
- **Strong identifiability:** The original signer and any third party can determine the identity of the actual proxy signer by checking the proxy signature.
- **Secret-keys' dependence:** A new proxy signature key is computed from the private key of an original signer. Furthermore, the original signer cannot calculate another one.
- **Strong undeniability:** Once a proxy signer generated a valid signature on behalf of the original signer, she/he cannot repudiate signature creation against the original signer.

So far, many threshold proxy signature schemes have been widely studied [15, 22, 24, 28, 48]. In a (t, n) threshold proxy signature scheme, which is a variant of the proxy signature scheme, the proxy signature key is shared among a group of n proxy signers delegated by the original signer. Any t or

more proxy signers can cooperatively sign messages on behalf of the original signer.

Based on Kim's scheme [28], Sun [48] proposed an efficient non-repudiable threshold proxy signature scheme with known signers. Sun's scheme is more efficient than the other threshold proxy signature schemes. The main advantage of Sun's scheme is that the verifier is able to identify the actual signers in the proxy group. However, the weakness of Sun's scheme is that it is vulnerable to the conspiracy attack [15, 22]. Hsu et al. [15] proposed a new non-repudiable proxy signature scheme with known signers that can withstand the conspiracy attack and is more efficient than Sun's scheme. Technically, the security of the above two non-repudiable threshold proxy signature schemes is based on the discrete logarithm problem [10].

In 2000, Wang et al. [53] proposed a new (t, n) threshold signature scheme with (k, l) threshold-shared verification. According to the security level of a document, not only the document can be signed by some specified signers in a group, but it can also be verified by some specified verifiers in another group. Later, quite some literatures were released [17, 50]. Lee et al. [30] proposed an untraceable (t, n) threshold signature scheme based on the Ohta-Okamoto signature scheme [39]. For the sake of privacy and safety, the identity of the signers should be anonymous in a democratic society. At the same time, their scheme can be extended to give the original signers the ability to prove they are true signers, and any t or more malicious participants cannot reconstruct the polynomial to derive other participants' private keys and system secrets.

In 2004, Tzeng et al. [51] proposed a novel variation of proxy signature scheme called threshold multi-proxy multi-signature scheme for a specified group of verifiers. It allows the group of original signers to delegate the signing capability to the designated group of proxy signers. Furthermore, a subset of verifiers in the designated verifier group can authenticate the proxy signature. Unfortunately, Bao et al. showed that Tzeng et al.'s scheme cannot resist frame attacks [3]. That is to say, after intercepting a valid proxy signature, an adversary can construct a new signature, which can be authenticated as if they were generated by the subset of the proxy group on behalf of the adversary. Hence, Bao et al. proposed an improvement to overcome the weaknesses of Tzeng et al.'s scheme. In 2007, Hsu et al. demonstrated that Tzeng et al.'s scheme is insecure [16]. Any verifier can check the validity of the proxy signature by himself/herself with no help of other verifies. Hence, Hsu et al. also proposed an improvement to eliminate the security leak. All previously proposed threshold multi-proxy multi-signature schemes for a specified group

of verifiers have been based on discrete logarithms.

In this article, we shall propose a new proxy signature scheme based on RSA problem and attempt to combine Lee et al.'s (t, n) untraceable scheme and the requirement of (k, l) threshold-shared verification. A new ((t_1, n_1), (t_2, n_2), (t_3, n_3)) proxy signature scheme shall be proposed where the verifiers are able to identify the actual signers in the original group and the proxy group.

3 The Proposed Scheme

In this section, we shall present a new ((t_1, n_1), (t_2, n_2), (t_3, n_3)) proxy signature scheme. The proposed scheme involves the following participants: the share distribution center (SDC), the original group (G_O), the proxy group (G_P) and the verifier group (G_V). Further, the signers in G_O and G_P should elect one of them as the clerks C_O and C_P. The services provided by SDC are to set system parameters, to manage the public directory and to initialize the scheme. Moreover, by generating the secret share with the assistance of the SDC, the computation and communication costs can be greatly reduced [15]. The services provided by clerk C_O for G_O are to collect and to construct a proxy share from the individual signatures generated by the original signers. Furthermore, C_O can verify the validity of these individual signatures from the actual original signers. Similarly, the services provided by clerk C_P for G_P are to collect and to construct a proxy signature from the individual proxy signatures generated by the proxy signers. Furthermore, C_P can verify the validity of these individual proxy signatures from 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. 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. Similarly, any t_3 or more out of n_3 verifiers ($1 \leq t_3 \leq n_3$) can represent G_V to verify the proxy signature.

Initially, SDC defines the system parameters as follows:

- p, q*:* two large primes. To ensure p and q are strong primes, let $p = 2p'+1$ and $q = 2q' + 1$, where p' and q' are also primes.
- N*:* the product of p and q, as $N = pq$.
- $\lambda(N)$*:* the double of the product of p' and q', as $\lambda(N) = 2p'q'$.
- L*:* a random number such that $GCD(L, \lambda(N)) = 1$ and $L \approx 10^{50}$.
- α*:* an element which is primitive in both $GF(p)$ and $GF(q)$.

- $h(\cdot)$*:* a one-way hash function.
- M_w*:* a warrant which records the identities of the original signers in G_O, the proxy signers in G_P and the verifiers in G_V, the parameters (t_1, n_1), (t_2, n_2), (t_3, n_3) 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 owns a private key $x_i = \alpha^{s_i} \bmod N$, where s_i is between 1 and $\lambda(N)$, and a public key

$$y_i = x_i^L \bmod N, \tag{1}$$

which is certified by a certificate authority (CA). Here, CA is responsible for issuing, management, and revoking certificates. Once CA has verified user's identity, it issues a user's public key certificate. The certificate involves some information that identifies himself/herself, such as his/her name and/or address.

SDC distributes private keys x_{O_i}, x_{P_j} and x_{V_k} to each original signer U_{O_i}, proxy signer U_{P_j} and verifier U_{V_k} ($i = 1, 2, ..., n_1; j = 1, 2, ...n_2; k = 1, 2, ..., n_3$). Let $G_O = \{U_{O_1}, U_{O_2}, \cdots, U_{O_{n_1}}\}$, $G_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{n_2}}\}$ and $G_V = \{U_{V_1}, U_{V_2}, \cdots, U_{V_{n_3}}\}$ be groups of n_1 original signers, n_2 proxy signers and n_3 verifiers, respectively.

The proposed scheme can be divided into four phases: the secret share generation phase, the proxy share generation phase, the proxy signature generation phase and the proxy signature verification phase. In the secret share generation phase, SDC generates shadows for all the proxy signers and verifiers in G_P and G_V. 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 verifiers cooperatively check the validity of the proxy signature and can identify not only the actual original signers, but also the actual proxy signers. The reference configurations and flow chart are illustrated in Figure 1. Details of the four stages are in the following subsections.

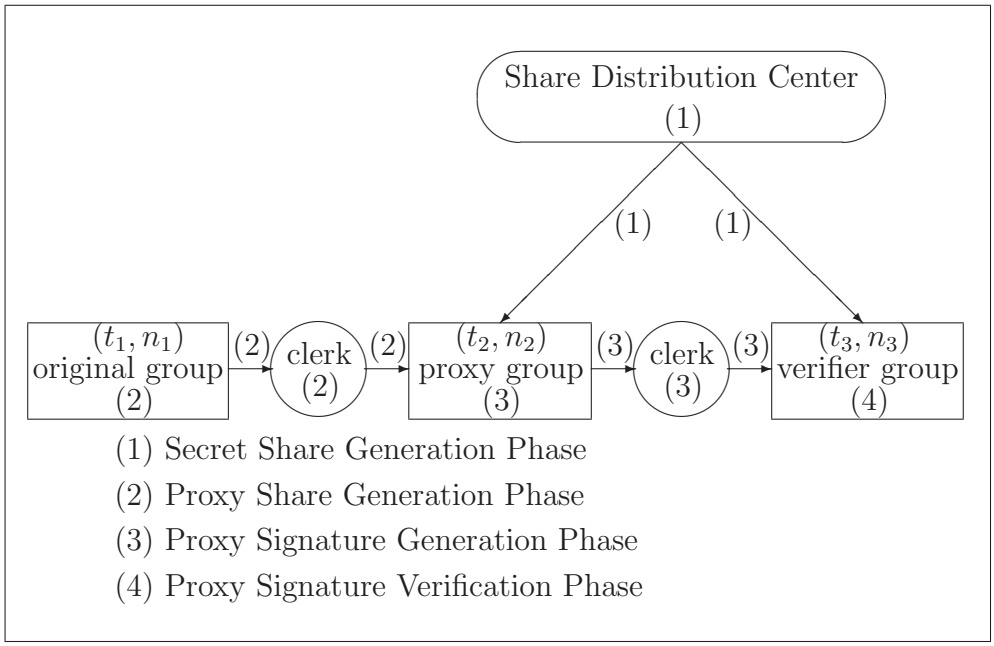

Figure 1: A Proxy Signature Scheme with Share Verification

3.1 Secret Share Generation Phase

The SDC executes the following steps to initialize parameters and sends shadows to G_P and G_V.

1. Calculate G_P's private key X_P and the corresponding G_P's public key Y_P, which is also certified by CA as follows:

$$\begin{cases} X_P &= \alpha^{-d} \bmod N, \\ Y_P &= X_P^L \bmod N, \end{cases}$$

   where d is a random number such that $GCD(d, \lambda(N)) = 1$. Once CA has verified G_P's identity, it issues G_P's public key certificate. Then, G_P's identity has been established.

2. Select two random numbers a and b which are between 1 and $\lambda(N)$ such that $GCD(a, b) = 1$. By using the extended Euclidean algorithm [42], given numbers a and b, there exist exactly two numbers c and w which are between 1 and $\lambda(N)$ and satisfy the equation $ac + bw = 1 \bmod \lambda(N)$.

3. Generate two secret polynomial functions $f_P(x)$ and $f_V(x)$ as

$$\begin{cases} f_P(x) &= dac + P_1 x + \cdots + P_{t_2-1} x^{t_2-1} \bmod \lambda(N), \\ f_V(x) &= dbw + V_1 x + \cdots + V_{t_3-1} x^{t_3-1} \bmod \lambda(N), \end{cases}$$

where the coefficients P_i and V_j are random numbers between 1 and $\lambda(N)$, $i = 1, 2, ..., (t_2 - 1)$ and $j = 1, 2, ..., (t_3 - 1)$. The degrees of the above polynomial functions $f_P(x)$ and $f_V(x)$ are $(t_2 - 1)$ and $(t_3 - 1)$, respectively.

4. Calculate the shadows x_{fPi} of the proxy signers in G_P and the shadows x_{fVj} of the verifiers in G_V as follows.

$$\begin{cases} x_{fPi} &= \alpha^{\frac{f_P(y_{P_i})/2}{(\prod_{k=1,k\neq i}^{n_2}(y_{P_i}-y_{P_k}))/2}} \bmod N, \\ x_{fVj} &= \alpha^{\frac{f_V(y_{V_j})/2}{(\prod_{k=1,k\neq j}^{n_3}(y_{V_j}-y_{V_k}))/2}} \bmod N, \end{cases}$$

where y_{P_i} and y_{V_j} are the public keys for U_{P_i} and U_{V_j}, respectively. The corresponding public information y_{fPi} and y_{fVj} of the above shadows x_{fPi} and x_{fVj} are also determined as

$$\begin{cases} y_{fPi} &= x_{fPi}^L \bmod N, \\ y_{fVj} &= x_{fVj}^L \bmod N. \end{cases}$$

The above parameters y_{fPi} and y_{fVj} are made public.

In summary, the parameters of the proposed scheme are listed in Table 1. The secret parameters in Table 1 have to be protected. However, the public parameters are accessible to anyone. The group entity consists of the proxy group and the verifier group.

Table 1: The parameters of the proposed scheme

Entity	Secret Parameters	Public Parameters
Share Distribution Center	$p, q, p', q', \alpha, \lambda(N)$	$N, L, h(\cdot)$
Original Signer	x_{O_i}	y_{O_i}
Proxy Signer	x_{P_i}, x_{fPi}	y_{P_i}, y_{fP_i}
Verifier	x_{V_i}, x_{fVi}	y_{V_i}, y_{fV_i}
Group	$a, b, c, d, w, X_P, f_P(x), f_V(x)$	Y_P

3.2 Proxy Share Generation Phase

The proposed scheme allows any t_1 or more original signers to represent G_O to delegate the signing capability to G_P to sign messages on behalf of G_O. Therefore, three parties, namely G_O, C_O and G_P, are involved in this phase. Without loss of generality, assume that the t_1' original signers to delegate the

signing capability are known as $D_O = \{U_{O_1}, U_{O_2}, \cdots, U_{O_{t_1'}}\}$, where $t_1 \leq t_1' \leq n_1$. Let D_O be the actual original signers. D_O is a group that executes the following steps to delegate the signing capability to G_P.

1. Each $U_{O_i} \in D_O$ selects a random number k_{O_i} between 1 and $(N-1)$ and calculates a value r_{O_i} as

$$r_{O_i} = k_{O_i}^L \bmod N. \tag{2}$$

   Then she/he broadcasts r_{O_i} to the other $(t_1' - 1)$ original signers in D_O and C_O.

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

$$K = \prod_{i=1}^{t_1'} r_{O_i} \bmod N, \tag{3}$$

$$\sigma_{O_i} = k_{O_i}^K x_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod N. \tag{4}$$

3. Each $U_{O_i} \in D_O$ broadcasts σ_{O_i} to C_O via a public channel.

4. After receiving these σ_{O_j} and r_{O_j}, C_O calculates K and checks whether the following equation holds:

$$\sigma_{O_i}^L \stackrel{?}{=} r_{O_i}^K y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod N. \tag{5}$$

   If it holds, $U_{O_i} \in D_O$ is a legal original signer and σ_{O_i} and r_{O_i} are the correct values. C_O can easily detect a wrong signature and then ask the actual original signer to send a valid one. Then, C_O can calculates σ_O as

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

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

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

$$\sigma_O^L \stackrel{?}{=} K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)} \bmod N. \tag{7}$$

If it does, each $U_{P_i} \in G_P$ accepts σ_O as a valid proxy share, and she/he calculates her/his proxy signature key σ_{P_i} as

$$\sigma_{P_i} = \sigma_O x_{P_i}^{y_{P_i}} \bmod N. \tag{8}$$

Otherwise, she/he rejects it and asks the original group of original signers for a valid one, or she/he stops this protocol.

3.3 Proxy Signature Generation Phase

Given a message M, the proposed scheme allows any t_2 or more proxy signers to represent G_P to sign M cooperatively on behalf of G_P. Therefore, G_P and C_P are involved in this phase. Without loss of generality, let $D_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{t_2'}}\}$ be the actual proxy signers, where $t_2 \leq t_2' \leq n_2$. D_P as a group executes the following steps for generating the proxy signature.

1. Each $U_{P_i} \in D_P$ selects a random number k_{P_i} between 1 and $(N-1)$ and calculates a value r_{P_i} as

$$r_{P_i} = k_{P_i}^L \bmod N.$$

   Then she/he broadcasts r_{P_i} to the other $(t_2' - 1)$ proxy signers in D_P.

2. Upon receiving these r_{P_j} $(j = 1, 2, \cdots, t_2'; j \neq i)$, each $U_{P_i} \in D_P$ calculates R and s_{P_i} as follows:

$$\begin{aligned} R &= \prod_{i=1}^{t_2'} r_{P_i} \bmod N, \\ s_{P_i} &= (x_{fPi}^{\prod_{k=t_2'+1}^{n_2}(y_{P_i}-y_{P_k})\prod_{k=1,k\neq i}^{t_2'}(0-y_{P_k})} k_{P_i})^R \sigma_{P_i}^{h(M,R,APSID)} \bmod N. \end{aligned} \quad (9)$$

   Here, s_{P_i} is the individual proxy signature which is sent to C_P via a public channel.

3. After receiving those s_{P_j}, C_P checks whether the following equation holds:

$$\begin{aligned} s_{P_i}^L \quad \stackrel{?}{=} \quad & (y_{fP_i}^{\prod_{k=t_2'+1}^{n_2}(y_{P_i}-y_{P_k})\prod_{k=1,k\neq i}^{t_2'}(0-y_{P_k})} r_{P_i})^R \times \\ & (K^K \prod_{j=1}^{t_1'} y_{O_j}^{y_{O_j} h(M_w,K,AOSID)} y_{P_i}^{y_{P_i}})^{h(M,R,APSID)} \bmod N. \end{aligned} \quad (10)$$

   If it does, then C_P calculates the value S as

$$S = \prod_{i=1}^{t_2'} s_{P_i} \bmod N. \quad (11)$$

   If it does not, C_P can easily determine that the individual proxy signature is wrong and then ask the actual proxy signer to send a valid one.

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

3.4 Proxy Signature Verification Phase

Any t_3 or more out of n_3 verifiers in G_V can cooperate to verify the validity of the proxy signature $(M_w, K, AOSID, M, R, S, APSID)$. Let $D_V = \{U_{V_1}, U_{V_2}, \cdots, U_{V_{t'_3}}\}$ be the actual verifiers, where $t_3 \leq t'_3 \leq n_3$. D_V as a group executes the following steps for verifying the validity of the proxy signature.

1. From M_w, $AOSID$ and $APSID$, each U_{V_j} gets the public keys of the original signers and proxy signers from CA and knows who the actual original signers and the actual proxy signers are.

2. Each $U_{V_i} \in D_V$ uses her/his shadow x_{fVj} to calculate and broadcast V_{V_j} as follows:

$$V_{V_j} = x_{fVj}^{R\prod_{k=t'_3+1}^{n_3}(y_{V_j}-y_{V_k})\prod_{k=1,k\neq j}^{t'_3}(0-y_{V_k})} \bmod N.$$

3. After receiving those V_{V_k} $(k = 1, 2, \cdots, t'_3; k \neq j)$, each $U_{V_j} \in D_V$ checks whether the following equation holds:

$$V_{V_j}^L = y_{fVj}^{R\prod_{k=t'_3+1}^{n_3}(y_{V_j}-y_{V_k})\prod_{k=1,k\neq j}^{t'_3}(0-y_{V_k})} \bmod N.$$

   If it does, each $U_{V_j} \in D_V$ calculates

$$V = \prod_{j=1}^{t'_3} V_{V_j} \bmod N. \tag{12}$$

4. Then, each $U_{V_j} \in D_V$ can check the validity of the proxy signature of M through the following equation:

$$\begin{aligned}(SV)^L Y_P^R \quad &\stackrel{?}{=} \quad R^R((K^K \prod_{i=1}^{t'_1} y_{O_i}^{y_{O_i}h(M_w,K,AOSID)})^{t'_2} \times \\ &\prod_{j=1}^{t'_2} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N. \end{aligned} \tag{13}$$

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

In the following analyses, we shall prove that the proposed scheme can work smoothly without fail.

Theorem 3.1 *In the proxy share generation phase, each $U_{P_i} \in G_P$ can verify the validity of σ_O sent by G_O by checking Equation (7).*

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

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

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

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

According to Equations (1), (2) and (3), the above equation can be rewritten as

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

$QED.$

Theorem 3.2 *If the proxy signature of the proposed scheme is constructed correctly, it will pass the verification of Equation (13).*

Proof. From Equations (9) and (11), we derive

$$\begin{aligned}S &= \prod_{i=1}^{t_2'} s_{P_i} \\ &= \prod_{i=1}^{t_2'} (x_{fPi}^{\prod_{k=t_2'+1}^{n_2} (y_{P_i} - y_{P_k}) \prod_{k=1, k\neq i}^{t_2'} (0 - y_{P_k})} k_{P_i})^R \sigma_{P_i}^{h(M,R,APSID)} \\ &= \prod_{i=1}^{t_2'} x_{fPi}^{R \prod_{k=t_2'+1}^{n_2} (y_{P_i} - y_{P_k}) \prod_{k=1, k\neq i}^{t_2'} (0 - y_{P_k})} \prod_{i=1}^{t_2'} k_{P_i}^R \sigma_{P_i}^{h(M,R,APSID)} \\ &= \alpha^{R \sum_{i=1}^{t_2'} f_P(y_{P_i}) \prod_{k=1, k\neq i}^{t_2'} \frac{0 - y_{P_k}}{y_{P_i} - y_{P_k}}} \prod_{i=1}^{t_2'} k_{P_i}^R \sigma_{P_i}^{h(M,R,APSID)} \bmod N.\end{aligned}$$

By using the Lagrange interpolating polynomial [42, 43], with the knowledge of t_2' pairs of $(y_{P_i}, f_P(y_{P_i}))$, the $(t_2 - 1)$th degree polynomial $f_P(x)$ can be determined as follows:

$$f_P(x) = \sum_{i=1}^{t_2'} f_P(y_{P_i}) \prod_{k=1, k\neq i}^{t_2'} \frac{x - y_{P_k}}{y_{P_i} - y_{P_k}} \bmod \lambda(N).$$

Thus,

$$\begin{aligned} S &= \alpha^{R\sum_{i=1}^{t'_2} f_P(y_{P_i}) \prod_{k=1,k\neq i}^{t'_2} \frac{0-y_{P_k}}{y_{P_i}-y_{P_k}}} \prod_{i=1}^{t'_2} k_{P_i}^R \sigma_{P_i}^{h(M,R,APSID)} \\ &= \alpha^{Rdac} \prod_{i=1}^{t'_2} k_{P_i}^R \sigma_{P_i}^{h(M,R,APSID)} \bmod N. \end{aligned}$$

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

$$\begin{aligned} S^L &= (\alpha^{Rdac} \prod_{i=1}^{t'_2} k_{P_i}^R \sigma_{P_i}^{h(M,R,APSID)})^L \\ &= \alpha^{LRdac} \prod_{i=1}^{t'_2} k_{P_i}^{LR} \sigma_{P_i}^{Lh(M,R,APSID)} \bmod N. \end{aligned}$$

For the same reason, by using the Lagrange interpolating polynomial, the value $f_V(0)$ can be obtained as follows:

$$f_V(0) = \sum_{j=1}^{t'_3} f_V(y_{V_j}) \prod_{k=1,k\neq j}^{t'_3} \frac{0-y_{V_k}}{y_{V_j}-y_{V_k}} \bmod \lambda(N).$$

Thus, Equation (12) can be rewritten as

$$\begin{aligned} V &= \prod_{j=1}^{t'_3} V_{V_j} \bmod N \\ &= \prod_{j=1}^{t'_3} x_{fVj}^{R\prod_{k=t'_3+1}^{n_3}(y_{V_j}-y_{V_k})\prod_{k=1,k\neq j}^{t'_3}(0-y_{V_k})} \bmod N \\ &= \alpha^{R\sum_{j=1}^{t'_3} f_V(y_{V_j}) \prod_{k=1,k\neq i}^{t'_3} \frac{0-y_{P_k}}{y_{P_j}-y_{P_k}}} \bmod N \\ &= \alpha^{Rdbw} \bmod N. \end{aligned}$$

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

$$V^L = \alpha^{LRdbw} \bmod N.$$

The left-hand side of Equation (13) can be rewritten as

$$\begin{aligned} (SV)^L Y_P^R &= S^L V^L Y_P^R \\ &= \alpha^{LRdac} \prod_{j=1}^{t'_2} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} \alpha^{LRdbw} Y_P^R \\ &= \alpha^{LRd(ac+bw)} \prod_{j=1}^{t'_2} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} Y_P^R \\ &= \alpha^{LRd} \prod_{j=1}^{t'_2} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} Y_P^R \end{aligned}$$

$$= \alpha^{LRd} \prod_{j=1}^{t_2'} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} \alpha^{-LRd}$$
$$= \prod_{j=1}^{t_2'} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} \bmod N.$$

Moreover, from Equations (1), (7) and (8), we get

$$\begin{aligned}
(SV)^L Y_P^R &= \prod_{j=1}^{t_2'} k_{P_j}^{LR} \sigma_{P_j}^{Lh(M,R,APSID)} \\
&= \prod_{j=1}^{t_2'} k_{P_j}^{LR} (\sigma_O x_{P_j}^{y_{P_j}})^{Lh(M,R,APSID)} \\
&= \prod_{j=1}^{t_2'} k_{P_j}^{LR} (\sigma_O^{t_2'} \prod_{j=1}^{t_2'} x_{P_j}^{y_{P_j}})^{Lh(M,R,APSID)} \\
&= R^R ((K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{t_2'} \prod_{j=1}^{t_2'} y_{P_j}^{y_{P_j}})^{h(M,R,APSID)} \bmod N.
\end{aligned}$$

Therefore, the correctness of proxy signature can be verified.

QED.

Theorem 3.3 *C_O can verify the validity of (r_{O_i}, σ_{O_i}) sent by U_{O_i} by checking Equation (5).*

Proof. Raising both sides of Equation (4) to exponents with L, we have

$$\begin{aligned}
\sigma_{O_i}^L &= (k_{O_i}^K x_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^L \\
&= k_{O_i}^{LK} x_{O_i}^{L y_{O_i} h(M_w,K,AOSID)} \bmod N.
\end{aligned}$$

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

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

Therefore, the correctness of individual proxy share can be verified.

QED.

Theorem 3.4 *C_P can verify the validity of (r_{P_i}, s_{P_i}) sent by U_{P_i} by checking Equation (10).*

Proof. According to Equation (8), we can rewrite Equation (9) as

$$s_{P_i} = (x_{fPi}^{\prod_{k=t_2'+1}^{n_2} (y_{P_i} - y_{P_k}) \prod_{k=1, k\neq i}^{t_2'} (0 - y_{P_k})} k_{P_i})^R (\sigma_O x_{P_i}^{y_{P_i}})^{h(M,R,APSID)} \bmod N.$$

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

$$\begin{aligned} s_{P_i}^L &= (x_{fPi}^{L\prod_{k=t_2'+1}^{n_2}(y_{P_i}-y_{P_k})\prod_{k=1,k\neq i}^{t_2'}(0-y_{P_k})} k_{P_i}^L)^R (\sigma_O^L x_{P_i}^{Ly_{P_i}})^{h(M,R,APSID)} \\ &= (y_{fPi}^{\prod_{k=t_2'+1}^{n_2}(y_{P_i}-y_{P_k})\prod_{k=1,k\neq i}^{t_2'}(0-y_{P_k})} r_{P_i})^R \times \\ &\quad (K^K \prod_{j=1}^{t_1'} y_{O_j}^{y_{O_j}h(M_w,K,AOSID)} y_{P_i}^{y_{P_i}})^{h(M,R,APSID)} \bmod N. \end{aligned}$$

Therefore, the correctness of individual proxy signature can be verified.

QED.

4 Discussions

In this section, the security analysis of the proposed scheme will be given first, followed by some discussions as to whether the proposed scheme can satisfy the requirements in [35, 36] a good proxy signature scheme should. In the next subsection, the performance analysis of our proposed scheme will be also given.

4.1 Security Analysis

The security of the proposed scheme is based on the well-known cryptographic assumptions:

1. **RSA problem [41]:**
   Let $N = pq$, where p and q are two large odd primes. There are two integers e and d satisfying $ed \equiv 1 \bmod \phi(N)$, where $\phi(N) = (p-1)(q-1)$ is an Euler's totient function. It is computationally infeasible to
   (a) find the factors of N;
   (b) find an integer M such that $M^e \equiv C \bmod N$, if C is known;
   (c) find an integer d such that $C^d \equiv M \bmod N$, if M and C are known.
2. $x^x = \beta \bmod N$ **problem [1, 12]:**
   Let N be a product of two large odd primes p and q. Given a fixed integer β in both $GF(p)$ and $GF(q)$, it is computationally infeasible to find x such that $x^x = \beta \bmod N$.

Next, we consider some possible attacks against the proposed scheme. Those attacks include trying to reveal the secret parameters [48], conspiring to derive the secret shadows [15, 22], and trying to forge the proxy share

or the proxy signature for a given warrant or message [15, 31, 48, 49]. Note that the attackers of the proposed scheme may come from insider (denoted as the original signer(s) or the proxy signer(s)) or outsider (denoted as adversary simple) of the group.

In the following, we analyze nine possible attacks to the proposed scheme. Attacks 1 and 2 concern that an adversary tries to derive the private keys from all available public information. Attacks 3 and 4 deal with the conspiracy attacks. Attack 3 concerns that any t_2 or more malicious proxy signers attempt to reconstruct the secret polynomial $f_P(x)$ and to derive the other participants' secret shadows. Attack 4 concerns that any t_3 or more malicious verifiers attempt to reconstruct the secret polynomial $f_V(x)$ and to derive the other participants' secret shadows. Attacks 5-9 deal with the forgery attacks. Attack 5 concerns that the malicious original signer tries to forge the proxy share for the arbitrary warrant. Attack 6 concerns that the malicious original signer attempts to forge a valid proxy signature for an arbitrary message. Attack 7 concerns that the malicious proxy signer tries to forge the proxy signature without the agreement of the other proxy signers. Attack 8 concerns that an adversary tires to forge a valid proxy share. Attack 9 concerns that an adversary attempts to forge a valid proxy signature. We shall prove that the proposed scheme can successfully withstand those possible attacks under the protection of the assumptions stated above.

Attack 1: An adversary attempts to derive a signer's or verifier's private key from all available public information.

Analysis of attack 1: Assume that the adversary wants to obtain U_{O_i}'s private key x_{O_i} from Equation (1). It is as difficult as breaking the RSA problem of N to derive U_{O_i}'s private key x_{O_i}. The adversary will face the intractability of the same problem in deriving U_{P_j}'s or U_{V_k}'s private key.

Attack 2: An adversary attempts to derive G_P's secret key X_P from all available public information.

Analysis of attack 2: As with Attack 1, the adversary will have to face the difficulty of solving the RSA problem of N to derive G_P's private key.

Attack 3: In G_P, any t_2 or more malicious proxy signers attempt to reconstruct the secret polynomial $f_P(x)$ of degree $(t_2 - 1)$ to derive the other participants' secret shadows x_{fPi}.

Analysis of attack 3: By using the Lagrange interpolating polynomial, with the knowledge of t_2 pairs of $(y_{P_i}, f_P(y_{P_i}))$'s, the (t_2-1)th degree polynomial $f_P(x)$ can be uniquely determined. Since SDC distributes $x_{fPi} = \alpha^{f_P(y_{P_i})} \bmod N$ instead of $f_P(y_{P_i})$ to G_P, any t_2 or more pairs of (y_{P_i}, x_{fPi}) cannot reconstruct the polynomial $f_P(x)$. The proxy signers need to find $f_P(y_{P_i})$ from x_{fPi}. It is as difficult as the problem of solving the discrete logarithm modulo a composite number N if α is known. Furthermore, the values α and $\lambda(N)$ are secret in the proposed scheme. Thus, no group of malicious proxy signers can collude to retrieve $f_P(x)$.

Attack 4: In G_V, any t_3 or more malicious verifiers attempt to reconstruct the secret polynomial $f_V(x)$ of degree (t_3-1) to derive the other participants' secret shadows x_{fVi}.

Analysis of attack 4: As with Attack 3, any t_3 or more malicious verifiers in G_V cannot reconstruct the polynomial $f_V(x)$ of degree (t_3-1) to obtain the other verifiers' secret shadows x_{fVi}.

Attack 5: Consider the insider forgery attack [31, 49] and direct forgery attack [49]. The 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 the arbitrary warrant.

Analysis of attack 5: 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 receives any (t_1-1) original signers' public keys y_{O_i} $(i = 2, 3, ..., t_1)$. She/he substitutes her/his public key y_{O_1}.

U_{O_1} selects a random number 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 as

$$y_{O_1}^{\prime y'_{O_1}} = x'^{L}_{O_1} \times (\prod_{i=2}^{t_1} y_{O_i}^{y_{O_i}})^{-1} \bmod N. \tag{14}$$

Finally, U_{O_1} selects a random number k_{O_1} and calculates K and σ_O as follows:

$$\begin{aligned} K &= k_{O_1}^L \bmod N, \\ \sigma_O &= k_{O_1}^K x_{O_1}^{\prime h(M_w, K, AOSID)} \bmod N. \end{aligned}$$

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

$$K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)}$$

$$
\begin{aligned}
&= K^K (y'^{y'_{O_1}}_{O_1} \prod_{i=2}^{t_1} y_{O_i}^{y_{O_i}})^{h(M_w,K,AOSID)} \\
&= K^K (x'^{L}_{O_1} \times (\prod_{i=2}^{t_1} y_{O_i}^{y_{O_i}})^{-1} \prod_{i=2}^{t_1} y_{O_i}^{y_{O_i}})^{h(M_w,K,AOSID)} \\
&= (k_{O_1}^K x'^{h(M_w,K,AOSID)}_{O_1})^L \\
&= \sigma_O^L \bmod N.
\end{aligned}
$$

We distinguish two cases here. In the first case, suppose U_{O_1} determines the value x'_{O_1} in Equation (14) first. She/he has to obtain the value y'_{O_1} by solving $x^x = \beta \bmod N$ problem. It is assumed to be computationally infeasible [1]. In the other case, suppose U_{O_1} wants to fix y'_{O_1}, she/he has to solve the RSA problem of N to find her/his public key x'_{O_1}. This is also assumed to be computationally infeasible. Therefore, the malicious original signer cannot successfully forge any proxy share for any warrant by launching the insider forgery attack.

Consider the direct forgery attack in [49]. U_{O_1} randomly selects a number r_{O_1} and calculates K and σ_O as

$$
\begin{aligned}
K^K &= r_{O_1}^L (\prod_{i=2}^{t_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{-1} \bmod N, \\
\sigma_O &= r_{O_1} x_{O_1}^{y_{O_1} h(M_w,K,AOSID)} \bmod N.
\end{aligned} \tag{15}
$$

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

$$
\begin{aligned}
& K^K \prod_{i=1}^{t_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \\
= \; & r_{O_1}^L (\prod_{i=2}^{t_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)})^{-1} y_{O_1}^{y_{O_1} h(M_w,K,AOSID)} \prod_{i=2}^{t_1} y_{O_i}^{y_{O_i} h(M_w,K,AOSID)} \\
= \; & (r_{O_1} x_{O_1}^{y_{O_1} h(M_w,K,AOSID)})^L \\
= \; & \sigma_O^L \bmod N.
\end{aligned}
$$

By the same arguments as above, U_{O_1} cannot calculate K to satisfy Equation (15). It is difficult to solve Equation (15) in polynomial time. Therefore, the malicious original signer in G_O cannot forge the proxy share by launching the direct forgery attack.

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

Analysis of attack 6: As with Attack 5, U_{O_k} should face the difficult problem or the RSA problem of N of generating a valid proxy signature. However, the value α^{LRdac} is secret. Therefore, the insider forgery attack by the malicious original signer in G_O cannot work.

Attack 7: 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 the insider forgery attack or the direct forgery attack.

Analysis of attack 7: U_{P_k} has to change her/his public key after the public keys of the other $(t_2 - 1)$ or more proxy signers have been determined. U_{P_k} has to face the same difficult problem as in Attacks 5 and 6. For the direct forgery attack as in Attack 5, the malicious proxy signer will face the intractability of the same problems to calculate R to forge the proxy signature. Furthermore, the value α^{LRdbw} is secret. Therefore, the malicious proxy signer in G_P cannot successfully forge any proxy signature for an arbitrary message by launching the insider forgery attack and the direct forgery attack.

Attack 8: An adversary attempts to forge a valid proxy share (σ_O, M_w, K, $AOSID$) of a chosen M_w to pass the proxy share verification equation.

Analysis of attack 8: First, we suppose

$$Y_O = \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i}} \bmod N.$$

She/he can rewrite Equation (7) as

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

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 RSA problem of N. Again, given M_w', $AOSID'$, σ_O' and Y_O, one cannot calculate a K' such that this equation holds, since it is assumed to be impossible to solve $x^x = \beta \bmod N$ problem in polynomial time. Therefore, the proxy share verification equation appears secure to against the forgery attack.

Attack 9: An adversary attempts to forge a valid proxy signature (M_w, K, $AOSID$, M, R, S, $APSID$) of some chosen M_w and M to pass the proxy signature verification equation.

Analysis of attack 9: First, we suppose

$$V_O = (K^K \prod_{i=1}^{t_1'} y_{O_i}^{y_{O_i} h(M_w, K, AOSID)})^{t_2'} \bmod N, \quad (16)$$

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

She/he can rewrite Equation (13) as

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

The value V_O depends on the parameters M_w, K and $AOSID$. V_P is a fixed value as the actual proxy signers' public keys are certified by CA. Given M', $APSID'$, V_O', V_P' and V', it is difficult to determine R' and S' because of the difficulty of solving the computationally infeasible problem and the RSA problem of N. Again, given M', $APSID'$, R', S', V_P' and V', one cannot calculate a V_O' such that the following equation holds:

$$V_O'^{h(M',R',APSID')} = (S'V')^L Y_P'^{R'} (R'^{R'} V_P'^{h(M',R',APSID')})^{-1} \bmod N.$$

The adversary has to obtain the value V_O' by solving the RSA problem of N. It is difficult to find M_w', $AOSID'$ and K' as a result of the value V_O' such that Equation (16) holds. Furthermore, the parameter V is secret. Only t_3 or more out of n_3 verifiers in G_V can cooperate to calculate the value V. Hence, the proxy signature verification equation appears secure against the forgery attack.

Next, we prove that the proposed proxy signature scheme is secure based on Random Oracle Model [9, 38, 46, 55, 56, 58]. Let A be an adversary. His/her goal is to forge a proxy signature using various ways. In order to forge a proxy signature, A can query two oracles, the random oracle and the signature oracle. At ths same time, there exists an adversary B who tries to break the hard RSA problem. The main idea of this proof is that B uses A as a tool to break the hard RSA problem. If we can show the fact that when A can forge a valid proxy signature with non-negligible probability, then B can break the hard problem with non-negligible probability. The main proof idea comes from Bellare and Rogaway's paper [4]. We denote t_{cost} as the main cost of reduction.

Theorem 4.1 *If the factoring problem is (π', ε')-secure, then for any q_{hash}, q_{sig} the proposed proxy signature scheme is $(\pi, q_{hash}, q_{sig}, \varepsilon)$-secure, where*

$$\pi = \pi' - (q_{sig} + q_{hash} + 1) \cdot t_{cost},$$
$$\varepsilon = (q_{sig} + q_{hash} + 1) \cdot \varepsilon'.$$

Proof. Let A be a forger, who can $(\pi, q_{hash},\ q_{sig}, \varepsilon)$-break the proposed proxy signature scheme. Then, we can use A to construct another B to solve factorization problem. B takes N, e, β as input and can compute x, satisfying $x^e \equiv \beta(\mathrm{mod}N)$ in π' steps and ε' probability where

$$\begin{aligned} \pi' &= \pi + (q_{sig} + q_{hash} + 1) \cdot t_{cost}, \\ \varepsilon' &= \varepsilon/(q_{sig} + q_{hash} + 1). \end{aligned}$$

B simulates a run of the proposed proxy signature scheme to the forge A. Furthermore, B should provides the hashing and signature oracle queries of A. For simplicity, we assume that if A makes sign m then it has already made hash oracle query m. We let $q = q_{sig} + q_{hash}$. B picks an integer j from $\{1, ..., q\}$ at random. Then, we can describe how B answers oracle queries. Here i is a counter, initially 0.

- Hashing oracle query: Suppose A makes hash oracle query m. B increments i and sets $m_i = m$. If $i = j$ then it sets $\beta_i = \beta$ and return β_i. Else it picks r_i at random in Z_N^*, sets $\beta_i \equiv r_i^e(\mathrm{mod}N)$, and returns β_i.

- Signature oracle query: Suppose A makes signing m. By assumption, there was already a hash query of m. so $m = m_i$ for some i. Let B return the corresponding signature.

A returns an attempted forgery signature and B outputs x. Without loss of generality, we assume that $m = m_i$ for some i. In this case, if the corresponding signature is valid forgery, then with at least the probability of $1/q$, we have $i = j$ and $x \equiv \beta_i^d \equiv \beta^d(\mathrm{mod}N)$.

The simulation shows that if we can break the proposed proxy signature scheme, we can break the the factoring problem, which of course is a contradiction for the RSA problem assumption. Therefore, we conclude that the assumption for the existence of B with non-negligible probability is invalid. Since there exists no B with non-negligible probability, it implies that no A can break our scheme. Therefore, based on factoring problem assumption, the proposed proxy signature scheme is secure. QED.

4.2 Our Requirements

In this subsection, we shall show the proposed scheme can satisfy the requirements mentioned in Section 2 as follows.

- *Strong unforgeability:* The original group can delegate their signing capability to a proxy group. Only a designated proxy group can generate a valid proxy signature on behalf of the original group. Any third party or even the original group cannot generate a valid proxy signature. According to the above Attacks 5-9, the proposed scheme meets the requirement of strong unforgeability.

- *Verifiability:* The proposed scheme can determine the identity information of the two parties since it appears in the proxy signature. In the proxy signature verification phase, the verifier group can make sure of the original group's agreement on the signed message. Hence, the proposed scheme provides the verifiability property.

- *Proxy signer's deviation:* A proxy group cannot generate a valid proxy signature not detected as their signature. The actual proxy signers' identity information is included in the signature. The proxy group cannot generate a valid ordinary signature of the original group according to the above security analysis, either. The property of proxy signer's deviation is fulfilled in the proposed scheme.

- *Distinguishability:* The proxy signatures and the self-signing signatures can be distinguished by anyone in polynomial time or size computation. It is mainly done by applying different congruence systems for the signature verification. That is to say, the proposed scheme satisfies the distinguishable property.

- *Strong identifiability:* The original signer or any third party can confirm from the proxy signature the identities of the corresponding proxy signers. Even the identities of the corresponding original signers can be determined. From $APSID$, we easily know 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. Thus, the proposed scheme achieves the strong identifiable property.

- *Secret-keys' dependence:* The individual proxy signature key is calculated by the proxy signer from a proxy share σ_O issued by the original group. Therefore, the proxy signature is dependent on the proxy share and proxy signers' private keys. In other words, the property of secret-key's dependence is present in the proposed scheme.

- *Strong un-deniability:* The actual proxy signers generate a valid proxy signature by using their private keys. The actual proxy signers cannot deny the creation of the proxy signature. Moreover, the proxy signature contains a warrant signed by the actual original signer's private key. Therefore, the original group cannot disavow it. As a result, the proposed scheme conforms to the property of strong un-deniability.

4.3 Performance Evaluation

In the following, the performance evaluation of the proposed scheme is discussed. We shall express the computational complexity and communication cost of the proposed scheme. We denote the performance evaluation notations as follows:

- T_{exp}: the time for a modular exponentiation computation.
- T_{mmul}: the time for a modular multiplication computation.
- T_{mul}: the time for a multiplication computation.
- T_{inv}: the time for a modular inverse computation.
- T_h: the time for a one-way hash function $h(\cdot)$ computation.
- $|x|$: the bit-length of an integer x.

The result of the computational complexity is given in Table 2 while the communication cost is shown in Table 3. In Table 2, the computation time for $x_{O_i}^{y_{O_i}}$, $x_{P_j}^{y_{P_j}}$, $y_{O_i}^{y_{O_i}}$ and $y_{P_j}^{y_{P_j}}$ is not calculate. The reason is that those items can be pre-computed. The computational complexities of executing the extended Euclidean algorithm and subtraction operations are neglected.

In the secret share generation phase, the SDC requires $2T_{exp} + T_{inv}$ computations in Step 1, $4T_{mmul}$ computations in Step 3, and $(2n_2 + 2n_3)T_{exp} + (n_2(n_2+t_2'-1)+n_3(n_3+t_3'-1))T_{mmul}+(n_2+n_3+1)T_{inv}$ computations in Step 4. Then, the total computations of the secret share generation phase adds up to be $(2n_2+2n_3+2)T_{exp}+(n_2(n_2+t_2'-1)+n_3(n_3+t_3'-1)+4)T_{mmul}+(n_2+n_3+2)T_{inv}$.

In the proxy share generation phase, each proxy signer requires T_{exp} computations in Step 1, $2T_{exp}+t_1'T_{mmul}+T_h$ computations in Step 2, and $(t_1'-1)T_{mmul}$ computations in Step 4. Then, the total computations of generating the proxy share is $3T_{exp} + (2t_1' - 1)T_{mmul} + T_h$. Afterwards, each proxy signer requires to generate her/his proxy signature key with $3T_{exp} + (t_1' + 1)T_{mmul} + T_h$ computations for Equations (7) and (8).

Table 2: The computational complexity of the proposed scheme

	Computations
Secret Share Generation Phase	$(2n_2 + 2n_3 + 2)T_{exp} + (n_2(n_2 + t'_2 - 1) + n_3(n_3 + t'_3 - 1) + 4)T_{mmul} + (n_2 + n_3 + 2)T_{inv}$
Proxy Share Generation Phase	Generating the proxy share: $3T_{exp} + (2t'_1 - 1)T_{mmul} + T_h$ Generating the proxy signature key: $3T_{exp} + (t'_1 + 1)T_{mmul} + T_h$
Proxy Signature Generation Phase	Generating the individual proxy signature: $4T_{exp} + (t'_2 + 1)T_{mmul} + (n_2 - 2)T_{mul} + T_h$ Generating the proxy signature: $(4t'_2 - 4)T_{exp} + (4t'_2 - 4)T_{mmul} + (t'_2 n_2 - n_2 - 2t'_2 + 2)T_{mul} + T_h$
Proxy Signature Verification Phase	$(2t'_3 + 6)T_{exp} + (t'_1 + t'_2 + t'_3 + 2)T_{mmul} + (t'_3 n_3 - t'_3)T_{mul} + 2T_h$

Table 3: The communication cost of the proposed scheme

	Communications
Secret Share Generation Phase	$(2n_2 + 2n_3 + 1)\|N\|$
Proxy Share Generation Phase	$2t'_1\|N\| + \|M_w\| + \|AOSID\|$
Proxy Signature Generation Phase	$(2t'_2 + 1)\|N\| + \|M\| + \|M_w\| + \|AOSID\| + \|APSID\|$
Proxy Signature Verification Phase	$(t'_3 - 1)\|N\|$

In the proxy signature generation phase, each proxy signer requires T_{exp} computations in Step 1 and $3T_{exp} + (t'_2 + 1)T_{mmul} + (n_2 - 2)T_{mul} + T_h$ computations in Step 2. The total computations for generating the individual proxy signature adds up to be $4T_{exp} + (t'_2 + 1)T_{mmul} + (n_2 - 2)T_{mul} + T_h$. The clerk C_P requires $(4t'_2 - 4)T_{exp} + (3t'_2 - 3)T_{mmul} + (t'_2 n_2 - n_2 - 2t'_2 + 2)T_{mul} + T_h$ and $(t'_2 - 1)T_{mmul}$ for checking the individual proxy signatures and generating the proxy signature, respectively. Therefore, the total computations for generating the proxy signature is $(4t'_2 - 4)T_{exp} + (4t'_2 - 4)T_{mmul} + (t'_2 n_2 - n_2 - 2t'_2 + 2)T_{mul} + T_h$.

In the proxy signature verification phase, each verifier requires $T_{exp} + (n_3 - 1)T_{mul}$ computations in Step 2, $2(t'_3 - 1)T_{exp} + (t'_3 - 1)T_{mmul} + (t'_3 - 1)(n_3 - 1)T_{mul}$ computations in Step 3, and $7T_{exp} + (t'_1 + t'_2 + 3)T_{mmul} + 2T_h$ computations in Step 4. The total computations of this phase is thus $(2t'_3 + 6)T_{exp} + (t'_1 + t'_2 + t'_3 + 2)T_{mmul} + (t'_3 n_3 - t'_3)T_{mul} + 2T_h$.

The communication cost of the proposed scheme is listed in Table 3. In the secret share generation phase, the communication cost is $|N|$ in Step 1, and $2(n_2 + n_3)|N|$ in Step 4. The total communications for the secret and public parameters are thus $(2n_2 + 2n_3 + 1)|N|$ in the secret share generation phase. In the proxy share generation phase, the communications required include $(t_1' - 1)|N|$ in Step 1, $(t_1' - 1)|N|$ in Step 3, and $2|N| + |M_w| + |AOSID|$ in Step 5. The total communications are thus $2t_1'|N| + |M_w| + |AOSID|$ in the proxy share generation phase. In the proxy signature generation phase, the communications of all the actual proxy signers are $(t_2' - 1)|N|$ in Step 1, $(t_2' - 1)|N|$ in Step 2, and $3|N| + |M| + |M_w| + |AOSID| + |APSID|$ in Step 3. The total communications of the proxy signature phase add up be $(2t_2' + 1)|N| + |M| + |M_w| + |AOSID| + |APSID|$. Finally, the total communications in the proxy signature verification phase are $(t_3' - 1)|N|$.

5 Conclusion

In this article, we have proposed a $((t_1, n_1), (t_2, n_2), (t_3, n_3))$ proxy signature scheme. The concepts of partial delegation with warrant, threshold delegation and shared verification are integrated. In the proposed scheme, any t_1 out of n_1 original signers in the original group can allow a designated proxy group to sign on behalf of the original group. Any t_2 out of n_2 proxy signers in the proxy group can use their own individual proxy signature keys and shadows to generate the proxy signature for a message. Meanwhile, any t_3 out of n_3 verifiers in the verifier group can cooperatively verify the proxy signature without releasing their shadows. Based on the difficulty of the RSA problem, the proposed scheme not only meets all the requirements a proxy signature should, but also provides desired security. 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. Moreover, the proposed scheme provides the ability to identify the actual original signers and actual proxy signers for avoiding the abuse of the signing capability.

References

[1] G. B. Agnew, B. C. Mullin, and S. A. Vanstone, "Improved digital signature scheme based on discrete exponentiation," *Electronics Letters*, vol. 26, no. 14, pp. 1024–1025, 1990.

[2] 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.

[3] H. Bao, Z. Cao, and S. Wang, "Improvement on Tzeng et al.'s nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Applied Mathematics and Computation*, vol. 169, no. 2, pp. 1419–1430, 2005.

[4] M. Bellare and P. Rogaway, "The exact security of digital signatures-how to sign with RSA and Rabin," in *Proceedings of Eurocrypt 1996*, pp. 399–416, Lecture Notes in Computer Science, 1996.

[5] K. Bicakci, "One-time proxy signatures revisited," *Computer Standards & Interfaces*, vol. 29, no. 4, pp. 499–505, 2007.

[6] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[7] S. Chang, D.S. Wong, Y. Mu, and Z. Zhang, "Certificateless threshold ring signature," *Information Sciences*, vol. 179, no. 20, pp. 3685–3696, 2009.

[8] S. W. Changchien, M. S. Hwang, and K. F. Hwang, "A batch verifying and detecting multiple RSA digital signatures," *International Journal of Computational and Numerical Analysis and Applications*, vol. 2, no. 3, pp. 303–307, 2002.

[9] H. R. Chung and W. C. Ku, "Three weaknesses in a simple three-party key exchange protocol," *Information Sciences*, vol. 178, no. 1, pp. 220–229, 2008.

[10] 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.

[11] 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.

[12] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," *IEE Proceedings - Computer and Digital Techniques*, vol. 141, no. 5, pp. 307–313, 1994.

[13] J. Herranz, "Identity-based ring signatures from RSA," *Theoretical Computer Science*, vol. 389, no. 1-2, pp. 100–117, 2007.

[14] X. Hong, "Efficient threshold proxy signature protocol for mobile agents," *Information Sciences*, vol. 179, no. 24, pp. 4243–4248, 2009.

[15] C. L. Hsu, T. S. Wu, and T. C. Wu, "New nonrepudiable threshold proxy signature scheme with known signers," *The Journal of Systems and Software*, vol. 58, no. 2, pp. 119–124, 2001.

[16] 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.

[17] C. L. Hsu, T. S. Wu, and T. C. Wu, "Improvements of threshold signature and authenticated encryption for group communications," *Information Processing Letters*, vol. 81, no. 1, pp. 41–45, 2002.

[18] 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.

[19] M. S. Hwang, C. C. Chang, and K. F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[20] M. S. Hwang and C. C. Lee, "Research issues and challenges for multiple digital signatures," *International Journal of Network Security*, vol. 1, no. 1, pp. 1–7, 2005.

[21] M. S. Hwang, C. C. Lee, and Y. C. Lai, "Traceability on RSA-based partially signature with low computation," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465-468, 2003.

[22] 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.

[23] M. S. Hwang, Eric J. L. Lu, and I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1–9, 2003.

[24] S. J. Hwang and C. C. Chen, "Cryptanalysis of nonrepudiable threshold proxy signature schemes with known signers," *Proceeding of* 12^{th} *National Conference on Information Security, R.O.C.*, pp. 243–246, 2002.

[25] S. J. Hwang and C. C. Chen, "New multi-proxy multi-signature schemes," *Applied Mathematics and Computation*, vol. 147, no. 1, pp. 57–67, 2004.

[26] S. J. Hwang and C. C. Chen, "New threshold-proxy threshold-signature schemes," *Computers and Electrical Engineering*, vol. 31, no. 1, pp. 69–80, 2005.

[27] Kitae Kim, Ikkwon Yie, and Seongan Lim, "Remark on Shao et al.'s bidirectional proxy re-signature scheme in Indocrypt'07," *International Journal of Network Security*, vol. 9, no. 1, pp. 8–11, 2009.

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

[29] F. Laguillaumie and D. Vergnaud, "Time-selective convertible undeniable signatures with short conversion receipts," *Information Sciences*, vol. 180, no. 12, pp. 2458–2475, 2010.

[30] N. Y. Lee, T. Hwang, and C. M. Li, "*(t, n)* threshold untraceable signatures," *Journal of Information Science and Engineering*, vol. 16, no. 6, pp. 835–845, 2000.

[31] 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.

[32] I. C. Lin and C. C. Chang, "Security enhancement for digital signature schemes with fault tolerance in RSA," *Information Sciences*, vol. 177, no. 19, pp. 4031–4039, 2007.

[33] Y. C. Liu, H. A. Wen, C. L. Lin, and T. Hwang, "Proxy-protected signature secure against the undelegated proxy signature attack," *Computers and Electrical Engineering*, vol. 33, no. 3, pp. 177–185, 2007.

[34] Chunbo Ma and Jun Ao, "Certificateless group oriented signature secure against key replacement attack," *International Journal of Network Security*, vol. 12, no. 1, pp. 1–6, 2011.

[35] 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.

[36] 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.

[37] Nikolay A. Moldovyan, "Blind signature protocols from digital signature standards," *International Journal of Network Security*, vol. 13, no. 1, pp. 22–30, 2011.

[38] J. Nam, Y. Lee, S. Kim, and D. Won, "Security weakness in a three-party pairing-based protocol for password authenticated key exchange," *Information Sciences*, vol. 177, no. 6, pp. 1364–1375, 2007.

[39] K. Ohta and T. Oktamoto, "A modification of the Fiat-Shamir scheme," in *Advances in Cryptology, CRYPTO'88*, pp. 232–243, Lecture Notes in Computer Science, 1988.

[40] 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. 7–12, 2011.

[41] 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.

[42] Kenneth H. Rosen, *Elementary Number Theory and Its Applications*. Addison Wesley, fourth edition, 2000.

[43] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, 1979.

[44] 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.

[45] Z. Shao, "Comment on improvement of the Miyazaki-Takaragi threshold digital signature scheme," *International Journal of Network Security*, vol. 3, no. 3, pp. 286–289, 2006.

[46] Z. Shao, "A provably secure short signature scheme based on discrete logarithms," *Information Sciences*, vol. 177, no. 23, pp. 5432–5440, 2007.

[47] Z. Shao, "Repairing efficient threshold group signature scheme," *International Journal of Network Security*, vol. 7, no. 2, pp. 218–222, 2008.

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

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

[50] Y. M. Tseng, J. K. Jan, and H. Y. Chien, "On the security of generalization of threshold signature and authenticated encryption," *IEICE Transactions on Fundamentals*, vol. E84-A, no. 10, pp. 2606–2609, 2001.

[51] S. F. Tzeng, C. Y. Yang, and M. S. Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Future Generation Computer Systems*, vol. 20, no. 5, pp. 887–893, 2004

[52] C. H. Wang and C. Y. Liu, "A new ring signature scheme with signer-admission property," *Information Sciences*, vol. 177, no. 3, pp. 747–754, 2007.

[53] C. T. Wang, C. C. Cheng, and C. H. Lin, "Generalization of threshold signature and authenticated encryption for group communications," *IEICE Transactions on Fundamentals*, vol. E83-A, no. 6, pp. 1228–1237, 2000.

[54] L. Wang, Z. Cao, X. Li, and H. Qian, "Simulatability and security of certificateless threshold signatures," *Information Sciences*, vol. 177, no. 6, pp. 1382–1394, 2007.

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

[56] Z. Wang, H. Qian, and Z. Li, "Hybrid proxy multisignature: A new type multi-party signature," *Information Sciences*, vol. 177, no. 24, pp. 5638–5650, 2007.

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

[58] A. Yasinsac and J. Childs, "Formal analysis of modern security protocols," *Information Sciences*, vol. 171, no. 1-3, pp. 189–211, 2005.

[59] 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.