On the Efficiency of Nonrepudiable Threshold Proxy Signature Scheme with Known Signers *

Cheng-Ying Yang[‡] Shiang-Feng Tzeng[†] Min-Shiang Hwang[†]

Department of Information Management[†]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-23742337

Graduate Institute of Networking and Communication Engineering[‡]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.

March 21, 2003

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-004.

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

On the Efficiency of Nonrepudiable Threshold Proxy Signature Scheme with Known Signers

Abstract

In the (t, n) proxy signature scheme, the signature signed by the original signer can be signed by t or more proxy signers out of a proxy group of n proxy signers. Recently, Hsu et al. proposed a nonrepudiable threshold proxy signature scheme with known signers. In this article, we shall propose an improvement of Hsu et al.'s scheme that is more efficient in terms of computational complexity and communication cost.

Keywords: Digital signature, Proxy signature, Threshold proxy signature.

1 Introduction

The concept of the proxy signature scheme was first introduced by Mambo et al. [8, 9] in 1996. A proxy signature scheme allows the original signer to delegate her/his signing capability to a designated person, called a proxy signer. The proxy signer on behalf of the original signer must act in such a way that the following requirements can be met:

1. Unforgeability. Besides the original signer, the designated proxy signer can also create a valid proxy signature on behalf of the original signer. Any third party who is not the designated proxy signer cannot create a valid proxy signature.

2. Verifiability. The verifier can be convinced of the original signer's agreement on the signed message because of the proxy signature.

So far, proxy signature schemes have been widely discussed [6, 7, 11, 12]. There are three types of delegation: full delegation, partial delegation, and delegation by warrant [8, 9]. In full delegation, the proxy signer is given the same private key as an original signer has. So, there is no distinguishing the proxy signer from the original signer by private key. In the partial delegation, the original signer uses her/his private key to create a proxy signature key and has it sent to the proxy signer. For security requirements, it is computationally infeasible for the proxy signer to derive 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. The last delegation uses a warrant which certifies the proxy signer delegated by the original signer.

According to the above description, from the angle of security, partial delegation and delegation by warrant are better than full delegation. Besides, compared with the delegation by warrant, partial delegation needs less processing time. Therefore, among these three types of delegation, partial delegation is the best choice. Therefore, throughout this article, we shall focus on the proxy signature authorized by partial delegation.

A digital signature scheme allows the signers to sign messages in such a way that everyone can verify the validity of the authentic signatures [2, 5]. However, nobody can forge the signatures of other messages. Hence the signer cannot repudiate the signatures which she/he has ever signed. This property is usually referred to as "nonrepudiation". However, in the proxy signature scheme, there exists a dispute as to partial delegation that, aside from the proxy signature key for the proxy signer, the original signer can derive another proxy signature key and save it for her/his own use to generate a valid proxy signature and impersonate the proxy signer. This problem can be solved using the nonrepudiation property, the capability of identifying the actual signer, to the proxy signature scheme.

Following the development of the proxy signature scheme, some threshold proxy signature schemes were widely studied [6, 12]. 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. So far, there have been at least two threshold proxy signature schemes proposed [6, 12]. Kim's scheme [6] is nonrepudiable, but Zhang's scheme [12] is not.

Based on Kim's scheme, Sun [11] proposed an efficient nonrepudiable threshold proxy signature scheme with known signers. Sun's scheme of nonrepudiability property 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 [3, 4]. Hsu et al. [3] proposed a new nonrepudiable proxy signature scheme with known signers. Hsu et al.'s scheme can withstand the conspiracy attack, and that is more efficient than Sun's scheme. The improved scheme proposed in this paper can also overcome any possible attacks such as plaintext attack, conspiracy attack and forgery attack. Furthermore, The improved scheme is better than Hsu et al.'s scheme in terms of computational complexity and communication cost.

In this paper, a brief review of Hsu et al.'s scheme will be given in the following section. Then, in Section 3, a secure scheme based on Hsu et al.'s scheme will be released. The security and the performance analysis of the proposed scheme will be discussed in Section 4. Finally, the conclusion will be given in Section 5.

2 Review of Hsu et al.'s Scheme

The scheme can be divided into four phases: secret share generation, proxy share generation, proxy signature generation, and proxy signature verification. Hsu et al.'s scheme is illustrated in Figure 2.

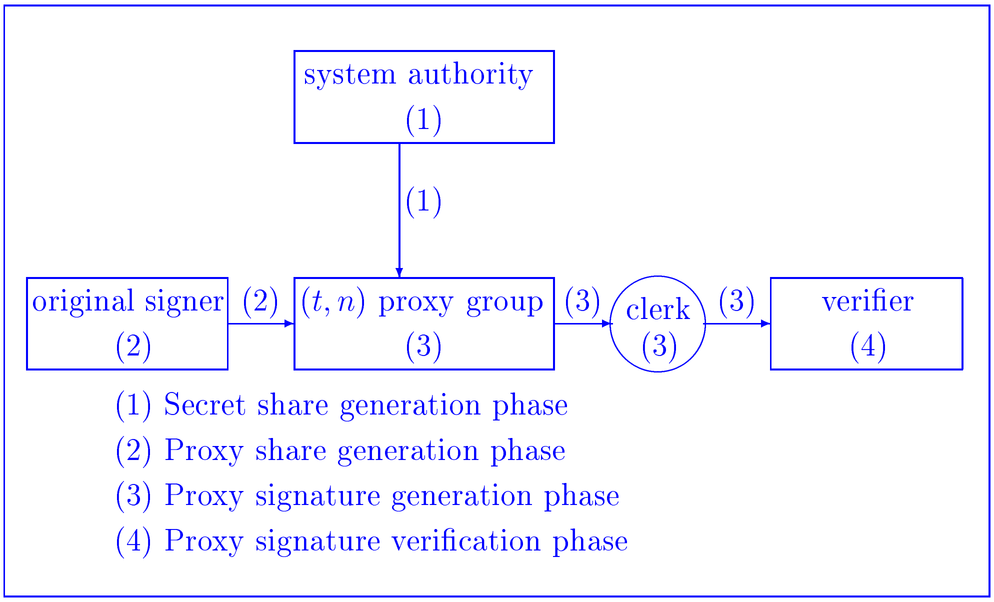

Figure 1: Hsu et al.'s scheme

There exists a system authority (SA) whose tasks are to initialize the system and manage the public directory. Initially, the SA selects and publishes the following parameters:

p: a large prime,

q: a large prime factor of $p-1$,

g: a generator in $GF(p)$ of order q,

$h(\cdot)$: a one-way hash function.

m_w: a warrant which records the identities of the original signer and the proxy signers of the proxy group, the parameters t and n, and the valid delegation time, etc.

$ASID$: (Actual Signers' ID) the identities of the actual signers.

Each user P_i, with the public identifier $v_i \in Z_q$, owns a private key $x_i \in Z_q^*$ and a public key $y_i = g^{x_i} \bmod p$ which is certified by a certificate authority (CA). Let P_0 be the original signer and $G = \{P_1, P_2, \cdots, P_n\}$ be the proxy group of n proxy signers.

2.1 Secret share generation phase

The SA chooses the group private key X_G and computes the group public key $Y_G = g^{X_G} \bmod p$ which is certified by the CA. Then, the SA randomly generates a $(t-1)$-degree polynomial as

$$f(v) = X_G + a_1 v + a_2 v^2 + \cdots + a_{t-1} v^{t-1} \bmod q$$

where the random integers $a_i \in Z_q$ $(i = 1, 2, \cdots, t-1)$.

For each $P_i \in G$, the SA computes the secret share $\gamma_i = f(v_i)$ and the corresponding public information $\tau_i = g^{\gamma_i} \bmod p$, where v_i is the public identifier for P_i. Then, the SA separately sends γ_i to P_i via a secure channel and publishes all τ_i's.

2.2 Proxy share generation phase

The original signer P_0 performs the following steps to delegate the signing capability to G.

1. Choose a random integer $k \in Z_q^*$ and compute $K = g^k \bmod p$.

2. Compute the proxy signature key as

$$\sigma = k + x_0 h(m_w \parallel K) \bmod q.$$

3. Choose a polynomial

$$f_0(v) = \sigma + b_1 v + b_2 v^2 + \cdots + b_{t-1} v^{t-1} \bmod q$$

where the random integers $b_j \in Z_q$ $(j = 1, 2, \cdots, t-1)$.

4. Publish

$$B_j = g^{b_j} \bmod p \quad for \;\; j = 1, 2, \cdots, t-1.$$

5. Send

$$\sigma_i = f_0(v_i)$$

to $P_i \in G$ via a secure channel.

6. Broadcast (m_w, K) to G.

Upon receiving σ_i, each $P_i \in G$ can validate it by checking the following equation

$$g^{\sigma_i} \stackrel{?}{=} y_0^{h(m_w \| K)} K (\prod_{j=1}^{t-1} B_j^{v_i^j}) \bmod p.$$

If it holds, P_i computes

$$\sigma_i' = \sigma_i + \gamma_i h(m_w \parallel K) \bmod q$$

as her/his proxy share.

2.3 Proxy signature generation phase

Given a message m, any t or more proxy signers of G will be the proxies for P_0 to sign m in this phase. Without loss of generality, let $D = \{P_1, P_2, \cdots, P_t\}$ be the actual proxy signers and $ASID$ (Actual Signers' ID) be the collection of identities of all the users in D. They sign m on behalf of P_0 cooperatively by performing the following steps.

1. Each $P_i \in D$ chooses a random integer $k_i \in Z_q^*$ and then broadcasts

$$r_i = g^{k_i} \bmod p.$$

2. Upon obtaining all r_j's ($j = 1, 2, \cdots, t$), each $P_i \in D$ computes

$$\begin{aligned} R &= \prod_{j=1}^{t} r_j \bmod p, \\ s_i &= k_i R + (L_i \sigma_i' + x_i) h(R \parallel ASID \parallel m) \bmod q, \end{aligned}$$

where $L_i = \prod_{j=1, j \neq i}^{t} (-v_j)(v_i - v_j)^{-1} \bmod q$. Here, s_i is the individual proxy signature which is sent to the designated clerk.

3. Upon receiving s_i, the designated clerk validates it by checking

$$g^{s_i} \stackrel{?}{=} r_i^R (((y_0 \tau_i)^{h(m_w \| K)} (\prod_{j=1}^{t-1} B_j^{v_i^j}) K)^{L_i} y_i)^{h(R \| ASID \| m)} \bmod p.$$

If it holds, (r_i, s_i) is the valid individual proxy signature of m. If all the individual proxy signatures of m are valid, the clerk computes

$$S = \sum_{j=1}^{t} s_j \bmod q.$$

The proxy signature of m is $(R, S, K, m_w, ASID)$.

2.4 Proxy signature verification phase

Receiving the proxy signature $(R, S, K, m_w, ASID)$ of m, the verifier can identify the original signer and the proxy group with the warrant m_w. Then, she/he knows the actual signers from $ASID$ and obtains the necessary public keys from the CA. The verifier can validate the proxy signature by checking

$$g^S \stackrel{?}{=} R^R (K (y_0 Y_G)^{h(m_w \| K)} \prod_{i=1}^{t} y_i)^{h(R \| ASID \| m)} \bmod p.$$

If it holds, the proxy signature $(R, S, K, m_w, ASID)$ for m is valid.

3 Improvement of Hsu et al.'s Scheme

The improved scheme is based on Hsu et al.'s scheme. The scheme can be divided into three phases: proxy share generation, proxy signature generation, and proxy signature verification. The architecture of the improved scheme is shown in Figure 2. In the improved scheme, SA doesn't need to generate secret shares and the corresponding public information for all the proxy signers. Therefore, as compared with Hsu et al.'s scheme, the secret share generate phase is not required. In the proxy share generation phase, the original signer generates the common proxy share and broadcasts it to the proxy group. Each proxy signer in the proxy group owns the same proxy share. In the proxy signature generation phase, the proxy signers cooperatively generate a valid proxy signature for a message on behalf of the original signer. In the proxy signature verification phase, the verifier checks the validity of the proxy signature. The system parameters p, q, g, $h(\cdot)$, m_w and $ASID$ are the same as those in Section 2. Each user P_i owns a private key $x_i \in Z_q^*$ and a public key

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

which is also certified by the CA. Let P_0 be the original signer and $G = \{P_1, P_2, \cdots, P_n\}$ be the proxy group of n proxy signers.

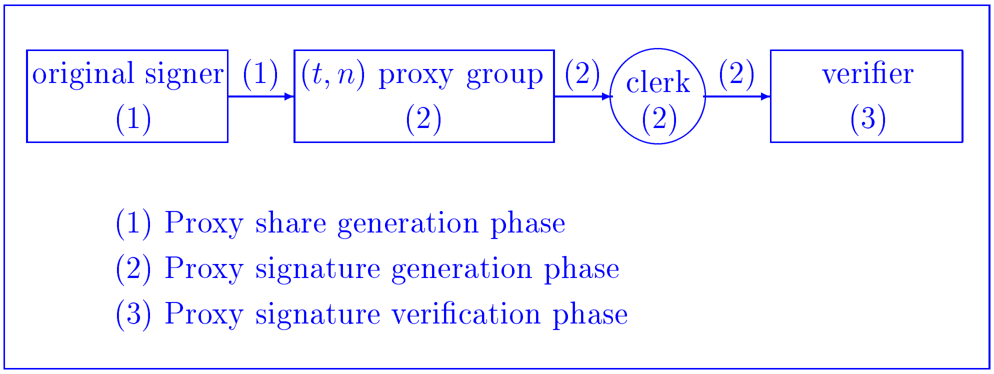

Figure 2: The proposed scheme

3.1 Proxy share generation phase

P_0 executes the following steps to delegate the signing capability to G.

1. Choose a random number $k \in Z_q^*$ and compute the parameter K.

$$K = g^k \bmod p. \tag{2}$$

2. Compute the proxy signature key as

$$\sigma = k + x_0 h(m_w \parallel K) \bmod q. \tag{3}$$

3. Broadcast (σ, m_w, K) to G.

After receiving (σ, m_w, K), each $P_i \in G$ checks whether or not the following equation holds,

$$g^\sigma \stackrel{?}{=} K y_0^{h(m_w \| K)} \bmod p. \tag{4}$$

If it holds, each P_i uses σ as her/his proxy share.

3.2 Proxy signature generation phase

Without loss of generality, we assume that any t or more proxy signers want to cooperate and sign a message m on behalf of the proxy group. Let $D = \{P_1, P_2, \cdots, P_t\}$ be the actual proxy signers.

1. Each $P_i \in D$ chooses a random number $k_i \in Z_q^*$ and broadcasts r_i.

$$r_i = g^{k_i} \bmod p. \tag{5}$$

2. For each received r_j $(j = 1, 2, \cdots, t; j \neq i)$, each $P_i \in D$ computes

$$R = \prod_{j=1}^{t} r_j \bmod p, \tag{6}$$

$$s_i = k_i R + (t^{-1}\sigma + x_i) h(R \parallel ASID \parallel m) \bmod q. \tag{7}$$

where t denotes the number of actual proxy signers. Here, s_i is the individual proxy signature which is sent to the designated clerk.

3. For each received s_i, the designated clerk checks whether the following equation holds,

$$g^{s_i} \stackrel{?}{=} r_i^R((Ky_0^{h(m_w\|K)})^{t^{-1}} y_i)^{h(R\|ASID\|m)} \bmod p. \tag{8}$$

If it holds, (r_i, s_i) is a valid individual proxy signature of m. If all the individual proxy signatures of m are valid, the designated clerk computes

$$S = \sum_{j=1}^{t} s_j \bmod q. \tag{9}$$

The proxy signature of m is $(R, S, K, m_w, ASID)$.

3.3 Proxy signature verification phase

Any verifier can confirm the validity of the proxy signature and identify the actual signers. The steps of this phase are described as follows:

1. According to m_w and $ASID$, the verifier gets the public keys of the original signer and the proxy signers from the CA and knows who are the original signer and the actual proxy signers.

2. The verifier has to check whether the number of the proxy signers in $ASID$ achieves or surpasses the threshold value t. A valid proxy signature can only be cooperatively generated by t or more proxy signers. Consequently, the proxy signature of creating by $t-1$ or less proxy signers is not a valid proxy signature, the verifier rejects it.

3. The verifier checks the validity of the proxy signature of the message m through the following equation:

$$g^S \stackrel{?}{=} R^R(Ky_0^{h(m_w\|K)} \prod_{j=1}^{t} y_j)^{h(R\|ASID\|m)} \bmod p. \tag{10}$$

If it holds, the proxy signature $(R, S, K, m_w, ASID)$ of m is valid.

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

Theorem 3.1 *In the proxy share generation phase, each $P_i \in G$ can verify the validity of the proxy share σ sent from P_0 by checking Equation (4).*

Proof. By raising both sides of Equation (3) to exponents with base g, we have

$$\begin{aligned} g^{\sigma} &= g^{k+x_0 h(m_w \| K)} \\ &= g^k g^{x_0 h(m_w \| K)} \bmod p. \end{aligned}$$

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

$$g^{\sigma} = K y_0^{h(m_w \| K)} \bmod p.$$

Hence, the correctness of proxy share can be confirmed. *QED.*

Theorem 3.2 *In the proxy signature generation phase, the designated clerk can verify the validity of individual proxy signature s_i sent from P_i by checking Equation (8).*

Proof. By raising both sides of Equation (7) to exponents with base g, we have

$$\begin{aligned} g^{s_i} &= g^{k_i R + (t^{-1}\sigma + x_i) h(R \| ASID \| m)} \\ &= g^{k_i R} (g^{t^{-1}\sigma} g^{x_i})^{h(R \| ASID \| m)} \bmod p. \end{aligned}$$

According to Equations (1), (4), and (5), the above equation can be rewritten as

$$g^{s_i} = r_i^R ((K y_0^{h(m_w \| K)})^{t^{-1}} y_i)^{h(R \| ASID \| m)} \bmod p.$$

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

Theorem 3.3 *In the proxy signature verification phase, any verifier can verify the validity of the proxy signature by checking Equation (10).*

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

$$g^S = \prod_{i=1}^{t} g^{s_i} \bmod p.$$

According to Equations (6) and (8), the above equation can be written as

$$\begin{aligned} g^S &= \prod_{i=1}^{t} r_i^R ((K y_0^{h(m_w\|K)})^{t^{-1}} y_i)^{h(R\|ASID\|m)} \\ &= \prod_{i=1}^{t} r_i^R \prod_{i=1}^{t} ((K y_0^{h(m_w\|K)})^{t^{-1}} y_i)^{h(R\|ASID\|m)} \\ &= R^R (K y_0^{h(m_w\|K)} \prod_{i=1}^{t} y_i)^{h(R\|ASID\|m)} \bmod p. \end{aligned}$$

Therefore, the correctness of proxy signature can be certified. *QED*.

4 Discusses

In this section, the security analysis of the improved scheme is given first and then the performance evaluation is also given.

4.1 Security Analysis

The security analysis of the improved scheme is similar to that of Hsu et al.'s scheme based on the following two well-known cryptographic assumptions:

1. **One-way hash function (OWHF) assumption [1, 10]:**

   Let $h(\cdot)$ be a one-way hash function. The function can take a message of arbitrary-length input and return a message digest of a fixed-length output. Given m, the function is easy to compute $h(m)$. However, the following tasks are computational infeasible:

   (a) Given $h(m)$, it is hard to derive m;

   (b) Given $h(m)$, it is hard to find $m' \neq m$ such that $h(m') = h(m)$.

2. **Discrete logarithm (DL) assumption [2, 10]:**

   Given a large prime p, a generator g of Z_p^* and an element $y \in Z_p^*$, it is computationally infeasible to find integer x such that $y = g^x \bmod p$.

In the following, let us discuss some possible attacks such as plaintext attack, conspiracy attack and forgery attack. The plaintext attack is performed by any adversary who attempts to derive the private keys from all available public information. The conspiracy attack concerns that the insider proxy signers attempt to obtain some other proxy signer's secret. In regard to the forgery attack the adversary tries to forgery a valid proxy signature. The detail process of above attacks is described as follows.

Plaintext attack

Assume the adversary attempts to derive the $P_0's$ private key x_0 from Equation (1). It is as difficult as breaking the DL assumption to derive user's private key x_0. Similarly, the adversary will meet the intractability of the same problem as obtaining $P_i's$ private key x_i for $i = 1, 2, \cdots, n$.

Conspiracy attack

Any t malicious proxy signers may work together to reconstruct the secret polynomial $f(v)$ in Section 2. Thus, they can conspire the secret share of any other proxy signer P_j. On the contrary, our improved scheme does not have the secret share generation phase. The security of both schemes as the above plaintext attack, they will have to face the difficulty of solving the DL assumption to derive $P_j's$ private key x_j in Hsu et al.'s scheme and the improved scheme. Therefore, the improved scheme can successfully with stand the conspiracy attack.

Forgery attack

An adversary tries to forge a valid proxy signature of some chosen m_w and m to pass the proxy signature verification equation. First we suppose

$$V_O = K y_0^{h(m_w \| K)} \bmod p. \tag{11}$$

Equation (10) can be rewritten as

$$g^S = R^R (V_O \prod_{j=1}^{t} y_j)^{h(R \| ASID \| m)} \bmod p.$$

The value V_O depends on the parameters m_w and K. We distinguish two cases here. In one case, give m', $ASID'$ and V'_O, it is difficult to determine R' and S' because of the difficulty of solving the OWHF and DL assumptions. In the other case, given m', $ASID'$, R' and S', one can calculate a V'_O such that this equation holds. However, it is difficult to find m'_w and K' such that Equation (11) holds. The difficulty here is also based on the OWHF and DL assumptions. Therefore, the proxy signature verification equation is secure against the forgery attack.

4.2 Performance Evaluation

In this subsection, we show the results of the comparisons between our improved scheme and Hsu et al.'s scheme in terms of computational complexity and communication cost. The performance evaluation notations are defined as follows:

T_{exp}: The time for a modular exponentiation computation.

T_{mul}: The time for a modular multiplication computation.

T_{inv}: The time for a modular inverse computation.

T_h: The time for computing a one-way hash function $h(\cdot)$.

$|\ x\ |$: The bit-length of an integer x.

The comparison results are given in Tables 1 and 2. As compared with Hsu et al.'s

Table 1: The comparison on computational complexity

	Hsu et al.'s scheme	The improved scheme
Secret share generation	$(n+1)T_{exp} + n(t-1)T_{mul}$	$\times$
Proxy share generation	The original signer: $tT_{exp} + (nt-n+1)T_{mul} + T_h$ Each proxy signer: $(t+1)T_{exp} + (2t-1)T_{mul} + T_h$	The original signer $T_{exp} + T_{mul} + T_h$ Each proxy signer: $2T_{exp} + T_{mul} + T_h$
Proxy signature generation	Individual proxy signature: $T_{exp} + (3t-1)T_{mul} + (t-1)T_{inv} + T_h$ Proxy signature: $(t^2+4t)T_{exp} + (4t^2-t-1)T_{mul} + (t^2-t)T_{inv} + 2T_h$	Individual proxy signature: $T_{exp} + (t+2)T_{mul} + T_{inv} + T_h$ Proxy signature: $(3t+2)T_{exp} + (2t+1)T_{mul} + T_{inv} + 2T_h$
Proxy signature verification	$4T_{exp} + (t+3)T_{mul} + 2T_h$	$4T_{exp} + (t+3)T_{mul} + 2T_h$

scheme, the secret shares are computed is not required. The computational complexities and communication costs of both schemes show that our improved scheme, needing no secret share generation phase, performs better than Hsu et al.'s scheme.

In Table 1, we can see the computational complexities of both schemes with the same proxy signature verification phase. The original signer generating a proxy share in the proxy share generation phase of the improved scheme, the computational complexity required includes T_{exp} for Equation (2) and $T_{mul} + T_h$ for Equation (3). The total computational complexity of the original signer is $T_{exp} + T_{mul} + T_h$. Each proxy signer checks the proxy share in the proxy share generation phase of the improved scheme, the computational complexity is $2T_{exp} + T_{mul} + T_h$ for Equation (4). However, the computational complexity of the original signer and each proxy signer in the same phase is $tT_{exp} + (nt-n+1)T_{mul} + T_h$ and $(t+1)T_{exp} + (2t-1)T_{mul} + T_h$ in Hsu et al.'s scheme. It is obvious that the improved scheme is more efficient than Hsu et al.'s scheme in this phase.

Table 2: The comparison on communication cost

	Hsu et al.'s scheme	The improved scheme
Secret share generation	$(n+1) \mid p \mid +n \mid q \mid$	$\times$
Proxy share generation	$t \mid p \mid +n \mid q \mid + \mid m_w \mid$	$\mid p \mid + \mid q \mid + \mid m_w \mid$
Proxy signature generation	$(t+1) \mid p \mid +t \mid q \mid + \mid m \mid + \mid m_w \mid$	$(t+1) \mid p \mid +t \mid q \mid + \mid m \mid + \mid m_w \mid$
Proxy signature verification	$2 \mid p \mid + \mid q \mid + \mid m \mid + \mid m_w \mid + \mid ASID \mid$	$2 \mid p \mid + \mid q \mid + \mid m \mid + \mid m_w \mid + \mid ASID \mid$

In the proxy signature generation phase of the improved scheme, the computational complexity that each proxy signer requires is T_{exp} for Equation (5) and $(t-1)T_{mul}$ and $3T_{mul} + T_{inv} + T_h$ for Equations (6) and (7) in Step 2. The total computational complexity of each proxy signer generating the individual proxy signature adds up to be $T_{exp} + (t+2)T_{mul} + T_{inv} + T_h$. The computational complexity for the designated clerk generating the proxy signature is $(3t+2)T_{exp} + (2t+1)T_{mul} + T_{inv} + 2T_h$ in the proxy signature generation phase of the improved scheme. However, the computational complexity for each proxy signer computing the individual proxy signature and the designated clerk generating the proxy signature is $T_{exp} + (3t-1)T_{mul} + (t-1)T_{inv} + T_h$ and $(t^2+4t)T_{exp} + (4t^2-t-1)T_{mul} + (t^2-t)T_{inv} + 2T_h$ in Hsu et al.'s scheme. The improved scheme is also more efficient than Hsu et al.'s scheme in the phase.

Table 2 shows the comparison on communication cost between Hsu et al.'s scheme and the improved scheme. The communication costs of both schemes are the same for the proxy signature generation and the proxy signature verification phases. In the proxy share generation phase of the improved scheme, the communication cost is $\mid p \mid + \mid q \mid + \mid m_w \mid$. However, the total communication cost of Hsu et al.'s scheme in the same phase is $t \mid p \mid +n \mid q \mid + \mid m_w \mid$. The improved scheme is also better than Hsu et al.'s scheme in this phase. Altogether, the comparisons show that the improved scheme is better than Hsu et al.'s scheme in terms of both computational complexity and communication cost.

5 Conclusions

In this article, we have presented an improved version of Hsu et al.'s threshold proxy signature scheme. The improved scheme has the same property that any t or more proxy signers may work together to generate a valid proxy signature on behalf of the original signer. The improved scheme also provides the ability to identity the actual proxy signers for avoiding the abuse of the signing capability. Moreover, the proxy signature contains a warrant signed by the original signer's private key. Therefore, the original signer cannot deny it. As a result, the improved scheme conforms the nonrepudiation property.

Some possible attacks such as plaintext attack, conspiracy attack and forgery attack have been considered. None of them can successfully break the improved scheme. As compared with Hsu et al.'s scheme, the secret shares are calculated is not required. Different from Hsu et al.'s scheme, the original signer only computes a common proxy share and broadcasts it to the proxy group. Therefore, the improved scheme is proven to be more efficient than Hsu et al.'s scheme in terms of computational complexity and communication cost.

References

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

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

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

[4] M. S. Hwang, I. C. Lin, and Eric J. L. Lu, "A secure nonrepudiable threshold proxy signature scheme with known signers," *International Journal of Informatica*, vol. 11, no. 2, pp. 1–8, 2000.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] Bruce Schneier, *Applied Cryptography, 2nd Edition.* New York: John Wiley & Sons, 1996.

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

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