A Threshold Signature Scheme for Group Communications without a Shared Distribution Center*

Chou-Chen Yang‡ Ting-Yi Chang‡ Min-Shiang Hwang†

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

Department of Information and Communication Engineering‡
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: ccyang@cyut.edu.tw

August 23, 2002

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

A Threshold Signature Scheme for Group Communications without a Shared Distribution Center

Abstract

In this paper, we shall propose a (t, n) threshold signature with (k, l) threshold-shared verification to be used in a group-oriented cryptosystem without a shared distribution center (SDC). In this scheme, any t participants can represent a group (signing group) to sign a message, and any k participants can represent another group (verifying group) to verify the signature. We need no SDC to distribute the public and private keys to all the participants in the two groups. Hence, our scheme is more practical in real-world applications and more efficient than its predecessors in terms of communication and computational complexity as well as storage.

Keywords: Cryptosystem, digital signatures, elliptic curves, secret sharing, threshold cryptosystem.

1 Introduction

Digital signatures have been played an important role in the modern electronic society. They have replaced a huge portion of the paperwork we used to count on by outperforming it with integrity and authentication. Along with the rapid advances in computer technology and the growth of the Internet, various types of digital signatures have been developed to live up to the requirements of our daily lives including business activities. Unlike such traditional digital signature schemes as RSA [1, 19] and DSA [14] where only a single signer is allowed to generate a signature for anyone to verify, threshold signature schemes allow t or more participants in the signer group to collaboratively generate a valid signature on behalf of the group, but only $t - 1$ or fewer participants will not be enough. Anyone can play the role of a verifier and check the correctness of the signature by using the group's public key.

The first (t, n) threshold signature scheme based on the RSA cryptosystem [19] and Shamir's secret sharing [21] was proposed by Desmedt and Frankel [2]. In 1994, Harn [5] combined a modified ElGamal signature scheme [21] and Shamir's secret sharing to accomplish a (t, n) threshold signature scheme. Later, several threshold signature schemes and their modifications where proposed in [6, 11] and others. On the other hand, in order to trace back to find the signers or to provide anonymity for the signers, several (t, n) threshold schemes with traceable or untraceable signers and their comments have been proposed in [10, 12, 23, 26] and other papers.

In 2000, Wang et al. [25] consider a situation where the documents between business entities need to be signed and verified. That is, the documents will not be exposed to any outsider. They bring up a new idea that the (t, n) threshold signature on behalf of the signing group should be able to be verified by (k, l) threshold shared verification on behalf of the verifying group. In their scheme [25], the shared distribution center (SDC) is responsible for dividing the signing group's and verifying group's secret keys into n and l different shadows and the associating the groups' and participants' public keys to the individual groups and participants, respectively. By using the Lagrange interpolation formula, t participants in the signing group and k participants in the verifying group have the ability to compute a common session key shared between two groups by using their shadows and the opposite group's public key. The common session key is used to ensure the communication between the two groups. Any t or more participants in the signing group can use their shadows to generate their individual signatures and hand over these individual signatures to a clerk. Then, the clerk can verify theses individual signatures and combine these t valid individual signatures to generate a threshold signature on behalf of the signing group. On the other hand, any k or more participants in the verifying group have the ability to collaborate to verify the threshold signature.

However, Tseng et al. [24] and Hsu et al. [7] have separately pointed that any adversary can reveal the signing group's secret key from two valid threshold signatures and then forge a threshold signature in scheme [25] because the common session key is always the same for different threshold signatures. It violates the basic definition requirement of the (t, n) threshold signature with (k, l) threshold shared verification. At the same time, Tseng et al. [24] and Hsu et al. [7] separately proposed their own improved versions of Wang et al's scheme [25]. However, though the common session key is changed for different threshold schemes in Tseng et al's improved scheme [24], the common session key is not separately computed by the signing group and the verifying group. In other words, the common session key generated by signing group should be sent to the verifying group. Anyone who obtains the session key can verify the threshold signature only by using the signing group's public key. This also violates the requirement upon the (t, n) threshold signature with (k, l) threshold shared verification. On the other hand, though Hsu et al's improved scheme [7] can successfully withstand the attack, their scheme has the following disadvantages in practice.

(1) The SDC is responsible for initializing the system and generating parameters.

(2) The SDC takes part in the generation of each threshold signature and the distribution of fresh shadows to all the participants.

(3) During the parameter generation phase performed by SDC, the distribution of the shadows is not verifiable against cheating by SDC.

(4) Each participant in the signing group should keep two private keys to sign a message.

(5) The signing group and the verifying group cannot exchange their roles with each other.

In 2001, Miyazaki and Takaragi [13] proposed another application for smart cards in threshold signature schemes. Consider $(2, 3)$ threshold signature schemes. A signer divides his/her private key into 3 pieces on his/her three smart cards. Then, he/she puts one of these cards in a strongbox as a kind of backup and usually uses the other two cards to sign a message. When one of the two usually used cards is lost, the method can prevent anyone who gets the lost card from forging the signer's signature, and the signer can issue his/her signature with the backup card. To keep within the restriction of storage and computation in a smart key, Miyazaki and Takaragi [13] realized a (t, n) threshold signature scheme based on the elliptic curve discrete logarithm problem (ECDLP). It is more efficient in view of communication and computational complexity as well as storage than previous schemes [16].

In this paper, we will modify Miyazaki and Takaragi's (t, n) threshold signature scheme and make it meet the requirement of (k, l) threshold shared verification. Because there is no SDC in our system, it is more practical and efficient in real-world applications than Hsu et al.'s scheme [7]. Moreover, our scheme can be extended so that the signature generated by the signing group can be verified by some special verifiers' collaboration in the verifying group.

The remainder of our paper is organized as follows. In Sections 2 and 3, we propose a new scheme and show the correctness of the proposed scheme, respectively. In Section 4, we shall analyze the security of our scheme. In Section 5, we shall compare the communication and computational complexity as well as the storage of our scheme with those of Hsu et al.'s scheme. In Section 6, there will be some discussions. Finally, we shall present our conclusion in Section 7.

2 Our Proposed Scheme

In order to give a clear picture, we begin with defining the following notations. The notation $G_s = \{u_{s1}, u_{s2}, \cdots, u_{sn}\}$ is defined as the signing group of n signers, and g_s ($|g_s| = t \leq n$) is any subset of size t in G_s. The notation $G_v = \{u_{v1}, u_{v2}, \cdots, u_{vl}\}$ is defined as the verifying group of l verifiers, and g_v ($|g_v| = k \leq l$) is any subset of size k in G_v. The notations ID_{si} and ID_{vi} are denoted as the identities of u_{si} and u_{vi}, respectively. Any t $u_{si} \in g_s$ can represent G_s, and any k $u_{vi} \in g_v$ can authenticate G_s's signature. The scheme is comprised of four phases: *(1) Key Generation Phase*, *(2) Individual Signature Generating and Verifying Phase*, *(3) Threshold Signature Generating and Encrypting Phase*, and *(4) Decrypting and Threshold Signature Verifying Phase*. Details of these phases will be stated in the below subsections.

2.1 Key Generation Phase

Pedersen's distributed key generation scheme [18] based on verifiable secret sharing [4, 17] is performed in this phase. Here, we shall separately show how each u_{si} in G_s and each u_{vi} in G_v to generate his/her private key, public key, and group public key. We will use the following notations:

E_s/E_v	Two elliptic curves.
p_s/p_v	Two odd prime numbers.
F_{p_s}/F_{p_v}	Finite fields of p_s and p_v elements, respectively.
α_s/α_v	Base points on E_s and E_v, respectively.
q_s/q_v	Orders of α_s and α_v separately in E_s and E_v, which are odd primes.

Each u_{si} in G_s performs the following steps:

Step 1. Randomly choose an integer d_{si}.

Step 2. Randomly choose a $(t-1)$th degree polynomial $f_{si}(x)$ over Z_{q_s} such that $f_{si}(x) = f_{si,0} + f_{si,1}x + \cdots + f_{si,t-1}x^{t-1}$, where $f_{si,0}, f_{si,1}, \cdots$, and $f_{si,t-1}$ are in Z_{q_s}. And $f_{si}(0) = f_{si,0} = d_{si}$. Then, send $f_{si}(ID_{sj})$ to

u_{sj} ($\forall j \neq i$) in G_s over a secret channel and broadcast the check values $f_{si,l} \cdot \alpha_s$ ($l = 1, 2, \cdots, t-1$) to all the other participants in G_s.

After receiving $f_{si}(ID_{sj})$ from u_{si}, each u_{sj} verifies the validity of it by the following verification equation:

$$f_{si}(ID_{sj}) \cdot \alpha_s \stackrel{?}{=} \sum_{l=0}^{t-1}(ID_{sj})^l \cdot (f_{si,l}\alpha_s). \tag{1}$$

If Equation (1) does not hold, reject u_{si}. Otherwise, each participant in G_s continues to perform the following steps.

Step 3. Compute his/her private key $K_{si} = \sum_{j=1}^{n} f_{sj}(ID_{si})$.

Step 4. Compute G_s's public key $Q_s = \sum_{j=1}^{n} f_{sj,0} \cdot \alpha_s$ and his/her public key $Q_{si} = K_{si} \cdot \alpha_s$.

Similarly, each u_{vi} in G_v performs the above steps. The result of performing those steps is listed in the following: $K_{vi} = \sum_{j=1}^{l} f_{vj}(ID_{vi})$ and $Q_{vi} = K_{vi} \cdot \alpha_v$ are separately u_{vi}'s private key and public key; And $Q_v = \sum_{j=1}^{l} f_{vj,0} \cdot \alpha_v$ is G_v's public key.

In summary, the system parameters are:

- Public information of G_s and G_v: E_s/E_v, α_s/α_v, Q_s/Q_v, q_s/q_v
- Public information of u_{si} in G_s and u_{vi} in G_v: Q_{si}/Q_{vi}, ID_{si}/ID_{vi}
- Secret information of u_{si} in G_s and u_{vi} in G_v: K_{si}/K_{vi}

2.2 Individual Signature Generating and Verifying Phase

According to our security policy, any t u_{si} in G_s can represent the signing group to sign a message based on the Nyberg-Rueppel signature scheme [15]. The participants can sign a message independently and simultaneously in this phase. Without loss of generality, assume that t participants $u_{s1}, u_{s2}, \cdots, u_{st}$ in g_s are to sign a message m. Each u_{si} performs the following steps.

Step 1. Compute a value e_{si} as

$$e_{si} = K_{si} \cdot a_{si}, \tag{2}$$

where $a_{si} = \prod_{\substack{j \in G_s \\ j \neq i}} \frac{ID_{sj}}{ID_{sj} - ID_{si}}$.

Step 2. Randomly choose an integer w_i, where $1 \leq w_i \leq q_s - 1$. Then, compute R_{si} as

$$R_{si} = w_i \cdot \alpha_s \tag{3}$$

and broadcast it to the other participants in g_s.

Step 3. Compute a point (X, Y) as

$$(X, Y) = \sum_{i \in g_s} R_{si} = \sum_{i \in g_s} w_i \cdot \alpha_s. \tag{4}$$

Step 4. Compute the individual signature $\{r, s_i\}$ as

$$r = X - h(m) \bmod q_s, \tag{5}$$

$$s_i = e_{si} \cdot r + w_i \bmod q_s. \tag{6}$$

To verify the correctness of the individual signature s_i, a participant may be randomly selected from G_s as a designated clerk. Except for verifying the individual signature, generating the threshold signature and encrypting the message, the clerk does not have any secret knowledge of the system.

Upon receiving the individual signature, the clerk uses u_{si}'s public key Q_{si} and a base point α_s to verify the individual signature as follows:

$$R_{si} \stackrel{?}{=} s_i \cdot \alpha_s - r \cdot a_{si} \cdot Q_{si}. \tag{7}$$

If Equation (7) holds, the individual signature $\{r, s_i\}$ on message m is valid.

2.3 Threshold Signature Generating and Encrypting Phase

In this phase, the clerk combines t valid individual signatures $\{r, s_i\}$ into a threshold signature $\{r, s\}$ and encrypts m by using the elliptic curve ElGamal cryptosystem [22] as follows:

Step 1. Compute the signature s as

$$s = \sum_{i \in g_s} s_i \bmod q_s. \tag{8}$$

$\{r, s\}$ is a group signature on message m.

Step 2. Express m as the x-coordinate of a point P_m on E_v [8]. Then, choose a random integer w_c, where $1 \leq w_c \leq q_v - 1$.

Step 3. Compute B and the ciphertext C as

$$\begin{aligned} B &= w_c \cdot \alpha_v \bmod q_v, \\ C &= P_m + w_c \cdot Q_v \bmod q_v. \end{aligned} \tag{9}$$

Step 4. Transfer $\{r, s\}$ and (B, C) to the verifying group G_v.

2.4 Decrypting and Threshold Signature Verifying Phase

To verify the signature $\{r, s\}$, any k u_{vi} in G_v can cooperate to decrypt the ciphertext C to obtain message m and authenticate the validity of the signature. Without loss of generality, assume that each of k participants $u_{v1}, u_{v2}, \cdots, u_{vk}$ in g_v wants to use his/her own private key K_{vi} to collaboratively recover the message and authenticate the signature by performing the following steps.

Step 1. Compute a value e_{vi} as

$$e_{vi} = B \cdot K_{vi} \cdot a_{vi}, \tag{10}$$

where $a_{vi} = \prod_{\substack{j \in G_v \\ j \neq i}} \frac{ID_{vj}}{ID_{vj} - ID_{vi}}$. Next, transfer e_{vi} to a clerk randomly selected from G_v.

Step 2. The clerk computes a point P_m as

$$P_m = C - \sum_{i \in g_v} e_{vi}, \tag{11}$$

and recover m from the x-coordinate of P_m.

Step 3. Compute $\hat{X}$-coordinate as

$$\hat{X} = r + h(m) \bmod q_s, \tag{12}$$

and compute the corresponding $\hat{Y}$-coordinate on E_s.

The signature can be verified by using the signing group's public key Q_s and the base point α_s as follows:

$$(\hat{X}, \hat{Y}) \stackrel{?}{=} s \cdot \alpha_s - r \cdot Q_s. \tag{13}$$

If Equation (13) holds, the signature $\{r, s\}$ on message m is valid.

3 The Correctness of Our Proposed Scheme

The correctness of the proposed scheme is shown in the following theorems.

Theorem 3.1 *Any participant u_{sj} in G_s can verify $f_{si}(ID_{sj})$ distributed by u_{si} in Equation (1).*

Proof. According to $f_{si,l} \cdot \alpha_s$ $(l = 1, 2, \cdots, t-1)$ broadcasted by u_{si}, Equation (1) can be rewritten as follows:

$$\begin{aligned} f_{si}(ID_{sj}) \cdot \alpha_s &= \sum_{l=0}^{t-1} (ID_{sj})^l \cdot (f_{si,l}\alpha_s), \\ &= (ID_{sj})^0 \cdot f_{si,0}\alpha_s + (ID_{sj})^1 \cdot f_{si,1}\alpha_s + \cdots + (ID_{sj})^{t-1} \cdot f_{si,t-1}\alpha_s. \end{aligned}$$

For the same reason, any participant u_{vj} in G_v can verify $f_{vi}(ID_{vj})$, which is distributed by u_{vi}. *Q.E.D.*

Theorem 3.2 *The message m encrypted in Equation (9) can be decrypted by k participants in G_s in Equation (11).*

Proof. According to Equations (9) and (10), we can rewrite Equation (11) as follows:

$$\begin{aligned} P_m &= C - \sum_{i \in g_v} e_{vi}, \\ &= C - \sum_{i \in g_v} B \cdot K_{vi} \cdot a_{vi}, \\ &= C - B \cdot (\sum_{i \in g_v} K_{vi} \cdot \prod_{\substack{j \in G_v, \\ j \neq i}} \frac{ID_{vj}}{,} ID_{vj} - ID_{vi}). \end{aligned}$$

By using the Lagrange formula, with the knowledge of k pairs of (ID_{vi}, K_{vi}), the unique d_v can be determined as follows.

$$d_v = \sum_{i \in g_v} K_{vi} \cdot \prod_{\substack{j \in G_v \\ j \neq i}} \frac{ID_{vj}}{ID_{vj} - ID_{vi}}.$$

Thus,

$$\begin{aligned} C - B \cdot (\sum_{i \in g_v} K_{vi} \cdot \prod_{\substack{j \in G_v \\ j \neq i}} \frac{ID_{vj}}{ID_{vj} - ID_{vi}}) &= C - B \cdot d_v, \\ &= C - w_c \cdot \alpha_v \cdot d_v, \\ &= P_m + w_c \cdot Q_v - w_c \cdot Q_v, \\ &= P_m. \end{aligned}$$

Then, m is represented by the x-coordinate of P_m. Therefore, the message can be recovered by k participants in G_v. *Q.E.D.*

Theorem 3.3 *The individual signatures can be verified by the clerk in Equation (7).*

Proof. According to Equation (3), we can rewrite the left-hand side of Equation (7) as follows:

$$R_{si} = w_i \cdot \alpha_s.$$

From Equation (6), the right-hand side of Equation (7) can be rewritten as follows:

$$\begin{aligned}
s_i \cdot \alpha_s - r \cdot a_{si} \cdot Q_{si} &= (e_{si} \cdot r + w_i) \cdot \alpha_s - r \cdot a_{si} \cdot K_{si} \cdot \alpha_s, \\
&= (K_{si} \cdot a_{si} \cdot r + w_i) \cdot \alpha_s - r \cdot a_{si} \cdot K_{si} \cdot \alpha_s, \\
&= w_i \cdot \alpha_s.
\end{aligned}$$

Therefore, the correctness of Equation (7) can be verified. *Q.E.D.*

Theorem 3.4 *The proposed scheme is a (t, n) threshold signature scheme with (k, l) threshold shared verification.*

Proof. According to Equations (2), (6) and (8), the signature s in Equation (13) can be rewritten as follows:

$$\begin{aligned}
s &= \sum_{i \in g_s} s_i \bmod q_s, \\
&= \sum_{i \in g_s} (e_{si} \cdot r + w_i) \bmod q_s, \\
&= \sum_{i \in g_s} K_{si} \cdot \prod_{\substack{j \in G_s \\ j \neq i}} \frac{ID_{sj}}{ID_{sj} - ID_{si}} \cdot r + w_i \bmod q_s.
\end{aligned}$$

By using the Lagrange formula,

$$\sum_{i \in g_s} K_{si} \cdot \prod_{\substack{j \in G_s \\ j \neq i}} \frac{ID_{sj}}{ID_{sj} - ID_{si}} \cdot r + w_i = d_s \cdot r + \sum_{i \in g_s} w_i \bmod q_s.$$

Thus, we can rewrite the right-hand side of Equation (13) as follows:

$$\begin{aligned}
s \cdot \alpha_s - r \cdot Q_s &= (d_s \cdot r + \sum_{i \in g_s} w_i) \cdot \alpha_s - r \cdot d_s \cdot \alpha_s, \\
&= \sum_{i \in g_s} w_i \cdot \alpha_s.
\end{aligned}$$

From Equation (4),

$$\begin{aligned}
\sum_{i \in g_s} w_i \cdot \alpha_s &= \sum_{i \in g_s} R_{si}, \\
&= (X, Y).
\end{aligned}$$

Because of Equation (5), we can obtain $\hat{X}$-coordinate in Equation (12) and the corresponding $\hat{Y}$-coordinate on E_s. Therefore, the correctness of Equation (13) can be verified. In this case, $\{r, s\}$ must be a signature generated by t signers in G_s. On the other hand, only k verifiers in G_v can cooperate to recover the message m (proved in Theorem 3.1), and then they have the ability to compute Equation (12) and verify the signature $\{r, s\}$ in Equation (13).

Q.E.D.

4 Security Analysis

The security level of the proposed (t, n) threshold signature scheme with (k, l) threshold shared verification is the same as that of Miyazaki and Takaragi's scheme [13], which is based on the intractability of the elliptic curve discrete logarithm problem (ECDLP). An adversary who intends to reveal a secret key from its corresponding public key will have to face ECDLP. In the rest of this section, several possible attacks will be raised and fought against to demonstrate the security of our scheme.

Attack 1: The participant u_{si} in G_s tries to distribute a fake $f_{si}(ID_{sj})'$ to u_{sj} $(\forall i \neq i)$ in G_s that can pass the verification of Equation (1).

Analysis of Attack 1: Obviously, if the check values $f_{si,l}\alpha_s$ $(l = 1, 2, \cdots, t-1)$ are announced to be genuine, the polynomial $f_{si}(x)$ remains the same as it was when generated before. Therefore, any fake $f_{si}(ID_{sj})'$ cannot successfully pass the verification of Equation (1). For the same reason, the participant u_{vi} in G_v also cannot distribute a fake $f_{vi}(ID_{vj})'$ to u_{vj} $(\forall i \neq i)$ in G_v.

Attack 2: An adversary tries to reveal G_s's secret key from the known public key Q_s.

Analysis of Attack 2: The difficulty is equivalent to solving ECDLP. Moreover, the polynomial $f_{si}(x)$ and the integer d_{si} are kept secret by u_{si}. Therefore, the adversary cannot reveal G_v's secret key from the known public key Q_v.

Attack 3: An adversary tries to reveal u_{si}'s secret key K_{si} from the known public key Q_{si}.

Analysis of Attack 3: As with *Attack 2*, the adversary will have to face the difficulty of solving ECDLP. Moreover, what u_{si} uses is a secret channel. Therefore, the adversary cannot reveal u_{vi}'s secret key K_{vi} from the known public key Q_{vi}.

Attack 4: An adversary tries to forge an individual signature to pass the verification of Equation (12) or forge a threshold signature to pass the verification of Equation (13).

Analysis of Attack 4: The security of the individual signatures and threshold signatures generated are provided by the Nyberg-Rueppel signature scheme [15]. If the adversary tries to forge an individual signature to pass the verification of Equation (12), he/she first chooses a random integer w_i' within $1 \leq w_i' \leq q_s - 1$ and broadcasts $R_{si}' = w_i'\alpha_s$. Without knowing u_{si}'s secret key K_{si}, the adversary will face the difficulty of generating a valid individual signature s_i' in Equation (6) to pass the verification of Equation (12). Furthermore, the adversary will obtain the sum point $(X,Y) = \sum_{\substack{j \in g_s \\ j \neq i}} R_{sj} + R_{si}'$. Without knowing u_{si}'s secret key K_{si}, the adversary will face the difficulty of generating a valid individual signature s_i' to satisfy the following equation:

$$(\hat{X},\hat{Y}) = (\sum_{\substack{j \in g_s \\ j \neq i}} s_j + s_i') \cdot \alpha_s - r \cdot Q_s.$$

On the other hand, without knowing G_s's secret key d_s, it is extremely difficult to forge a threshold signature to pass the verification of Equation (13).

Attack 5: An adversary tries to recover m from (B, C).

Analysis of Attack 5: The difficulty is equivalent to breaking the elliptic curve ElGamal cryptosystem.

5 Comparisons

In this section, we shall compare the communication and computational complexity as well as storage of our scheme with those of Hsu et al.'s scheme [7] based on discrete logarithm problem (DLP). In their scheme, SDC first prepares three polynomials. Two of these polynomials are with $(t-1)$ degree and for generating group signatures. The other is with $(k-1)$ degree and for verifying group signatures. In our scheme, after executing key generation phase, only one polynomial is separately used in the signing and verifying group signatures. Moreover, the public and private parameters distributed by SDC are not verifiable in their scheme (See [7] for the detail). Hence, we compare the numbers of public and private parameters held by each participant in the signing group and verifying group, respectively, after the key generation phase.

Table 1: The number of parameters held by each participant

	u_{si} in G_s		u_{vi} in G_v	
	Public values	Private values	Public values	Private values
Hsu et al.'s scheme	3	2	2	1
Our scheme	2	1	2	1

From Table 1, we learn that the numbers of public and private parameters held by u_{si} in G_s in our scheme are smaller than those in Hsu et al.'s scheme. Furthermore, in Hsu et al.'s scheme, one of the private parameters kept by u_{si} in G_s has to be redistributed from SDC over a secret channel for generating each threshold signature. The associated public parameter of that secret parameter will be republished. Due to fact that each participant in the signing group and verifying group has different amount of private keys, the signing group and the verifying group cannot exchange their roles with each other.

An elliptic curve $E(F_p)$ with a point $\alpha \in E(F_p)$ whose order is a 160-bit prime offers approximately the same level of security as DSA with a 1024-bit modulus p [9]. Hsu et al. employ the modified ElGamal signature scheme [3], and we assume that the modulus P is around 1024-bit in their scheme. To analyze the computation complexity, we first define the following notations. T_{EC}: the time for computing $k\alpha$; $T_{ElGamal}$: the time for computing $a^k \bmod p$. The authors of [9, 20, 27] have pointed out that computing $k\alpha$ requires an average of 29 1024-bit modular multiplications and computing $a^k \bmod p$ by doing repeated multiplications requires an average of 240 1024-bit modular multiplications. Thus, computing $k\alpha$ can be expected to be about 8 times faster than computing $a^k \bmod p$, i.e., $8T_{EC} = T_{ElGamal}$.

Table 2: Computational complexities of Hsu et al.'s scheme and our scheme

	Individual signature generating and verifying	Group signature generating and verifying
Hsu et al.'s scheme	$3t\ T_{ElGamal}$	$t+3\ T_{ElGamal}$
Our scheme	$4t\ T_{EC}$ for computing Equations (2), (3) and (7)	$t+4\ T_{EC}$ for computing Equations (9), (10) and (13)

According to Table 2, it is obvious that our scheme is more efficient than Hsu et al.'s scheme. Though we have to encrypt the message, our scheme is still faster by $20t$ and $7t+20$ 1024-bit modular multiplications than Hsu et al.'s scheme in the individual signature generating and verifying process and in the group signature generating and verifying process, respectively. On the other hand, consider the pre-computations. In Hsu et al.'s scheme, each participant in G_s should wait for a fresh secret parameter to generate individual signature, and each participant in G_v should wait for a fresh public parameter to verify a threshold signature. Table 3 shows that our scheme is much more efficient than Hsu et al.'s scheme with respect to pre-computations.

Table 3: Computational complexities of Hsu et al.'s scheme and our scheme with respect to pre-computations

	Individual signature generating and verifying	Group signature generating and verifying
Hsu et al.'s scheme	$6t\ T_{ElGamal}$	$t+3\ T_{ElGamal}$
Our scheme	$2\ T_{EC}$ for computing Equations (7)	$4\ T_{EC}$ for computing Equations (9) and (13)

An elliptic curve $E(F_p)$ with a point $\alpha \in E(F_p)$ whose order is a 160-bit prime offers approximately the same level of security as the modified ElGamal signature scheme [3] with a 1024-bit modulus P, and the prime divisor q of $P-1$ is 160-bit in Hsu et al.'s scheme. Here, we compare the data transferred inside the group and between two groups in our scheme and in Hsu et al.'s scheme. The notations are defined as follows. $u_{si} \longmapsto u_{sj}$: the communication from u_{si} to u_{sj} $(i \neq j)$ in g_s; $u_{si} \longmapsto$clerk: the communication from u_{si} to clerk in G_s (Assume that the clerk is not selected in g_s); $G_s \longmapsto G_v$: the communication from G_s to G_v; $u_{vi} \longmapsto$clerk: the communication from u_{vi} to clerk in G_v.

Table 4: The communications of Hsu et al.'s scheme and our scheme

	$u_{si} \longmapsto u_{sj}$	$u_{si} \longmapsto$clerk	$G_s \longmapsto G_v$	$u_{vi} \longmapsto$clerk
Hsu et al.'s scheme	$t-1 \times 1024$-bit	$3t \times 160$-bit	2×160-bit	$k \times 1024$-bit
Our scheme	$t-1 \times 160$-bit	$3t \times 160$-bit	4×160-bit	$k \times 160$-bit

According to Table 4, it is obvious that the total communication load in our scheme is less than that in Hsu et al.'s scheme. However, the communication demand between G_s and G_v in our scheme is 2 times larger than that in Hsu et al.'s scheme because there is no SDC to distribute the session keys between two groups in advance in our scheme. Our scheme needs to transfer the signature $\{r, s\}$ and encrypted message (B, C) to the verifying group.

6 Discussions

In this section, we shall discuss some special cases in our scheme will probably encounter. First, if the clerk selected from the signing group does not encrypt the message by using the verifying group's public key, anyone can play the role of a verifier to verify the signature. For the same reason, the clerk can encrypt the message by using the public keys of some special participants in the verifying group, and then only those participants can collaboratively recover the message and then verify the signature. For example, suppose the clerk encrypts the message m by using u_{v1}'s and u_{v2}'s public keys in Equation (9) as follows:

$$C = P_m + w_c \cdot (Q_{v1} + Q_{v2}) \bmod q_v.$$

Then, u_{v1} and u_{v2} separately compute e_{v1} and e_{v2} as follows:

$$\begin{aligned} e_{v1} &= B \cdot K_{v1} \\ e_{v2} &= B \cdot K_{v2}. \end{aligned}$$

The message m can be recovered by the following Equation (11).

$$\begin{aligned} P_m &= C - \sum_{i=1}^{2} e_{vi} \\ &= P_m + w_c \cdot (Q_{v1} + Q_{v2}) - B \cdot (K_{v1} + K_{v_2}), \\ &= P_m + w_c \cdot (Q_{v1} + Q_{v2}) - w_c \cdot \alpha_v \cdot (K_{v1} + K_{v_2}), \\ &= P_m. \end{aligned}$$

After the recovery of the message m, the signature $\{r, s\}$ can be verified. In the above description, our scheme provides another type of (k, l) threshold shared verification.

On the other hand, the participants in the signing group and those in the verifying group employ different system parameters (elliptic curves, base points, etc.) in our proposed scheme. In fact, each participant in the signing group and the verifying group can employ the same system parameters to perform four phases in our scheme. It does not harm the security of our scheme.

7 Conclusion

In this paper, we have added the requirement of (k, l) threshold shared verification to Miyazaki and Takaragi's scheme, and the security of our new scheme is based on the ECDLP. In addition, the communication and computational complexity as well as storage of our scheme turn out superior to those of Hsu et al.'s scheme. Besides, without the SDC, our scheme is more practical in real-world applications.

References

[1] Chin-Chen Chang and Min-Shiang Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[2] Y. Desmedt and Y. Frankel, "Shared generation of authenticators," in *Advances in Cryptology, CRYPTO'91*, pp. 457–469, 1991.

[3] 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.

[4] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. 28th IEEE Symp. FOCS*, pp. 427–437, 1987.

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

[6] P. Hoster, M. Michels, and H. Peterson, "Comment: Digital signature with (t, n) shared verification based on discrete logarithms," *IEE Electronics Letters*, vol. 31, no. 14, p. 1137, 1995.

[7] Chien-Lung Hsu, Tzong-Sun Wu, and Tzong-Chen Wu, "Improvements of threshold signature and authenticated encryption for group communications," *Information Processing Letters*, vol. 81, no. 1, pp. 41–45, 2002.

[8] Neal Koblitz, "Algebraic aspects of cryptography," *Algorithms and Computation in Mathematics*, vol. 3, 1998.

[9] Neal Koblitz, Alfred Menezes, and Scott A. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 9, no. 2/3, pp. 173–193, 2000.

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

[11] W. B. Lee and C. C. Chang, "Comment: Digital signature with *(t, n)* shared verifications based on discrete logarithms," *IEE Electronics Letters*, vol. 31, no. 19, pp. 1656–1657, 1995.

[12] Z. C. Li, L. C. K. Hui, K. P. Chow, C. F. Chong, W. W. Tsang, and H. W. Chan, "Security of wang et al.'s group-oriented *(t, n)* threshold signature schemes with traceable signers," *Information Processing Letters*, vol. 80, no. 6, pp. 295–298, 2001.

[13] Kunihiko Miyazaki and Kazuo Takaragi, "A threshold digital signature scheme for a smart card based system," *IEICE Transactions on Fundamentals*, vol. E84-A, no. 1, pp. 205–213, 2001.

[14] National Institute of Satndards and Technology (NIST), "The digital sinature standard proposed by NIST," *Communications of the ACM*, vol. 35, no. 7, pp. 36–40, 1992.

[15] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the DSA giving message recovery," in *1st ACM Conference on Computer and Communications Security*, pp. 58–61, Fairfax, Virginia, Nov. 1993.

[16] C. Park and K. Kurosawa, "New ElGamal type threshold digital signature scheme," *IEICE Transactions on Fundamentals*, vol. E79-A, no. 1, pp. 86–93, 1996.

[17] T. P. Pedersen, "Non-interactive and information-theoretic verifiable secret sharing," in *Advances in Cryptology, CRYPTO'91*, pp. 129–140, 1991.

[18] T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology, CRYPTO'91*, pp. 522–526, 1991.

[19] 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.

[20] R. Schroeppel, H. Orman, S. O'Malley, and O. Spatscheck, "Fast key exchang with elliptic curve systems," in *Advances in Cryptology, CRYPTO'95*, pp. 43–56, 1995.

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

[22] Wade Trappe and Lawrence C. Washington, *Introduction to Cryptography with Coding Theory.* Prentice Hall, 2002.

[23] Yuh-Min Tseng and Jinn-Ke Jan, "Attacks on threshold signature schemes with traceable signers," *Information Processing Letters*, vol. 71, no. 1, pp. 1–4, 1999.

[24] Yuh-Min Tseng, Jinn-Ke Jan, and Hung-Yu 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.

[25] Ching-Te Wang, Chin-Chen Chang, and Chu-Hsing Lin, "Generalization of threshold signature and authenticated encryption for group communications," *IEICE Transactions on Fundamentals*, vol. E83-A, no. 6, pp. 1228–1237, 2000.

[26] Ching-Te Wang, Chu-Hsing Lin, and Chin-Chen Chang, "Research note threshold signature schemes with traceable signers in group communications," *Computer Communications*, vol. 21, no. 8, pp. 771–776, 1998.

[27] E. De Win, A. Bosselaers, S. Vandenberghe, P. De Gersem, and J. Vandewalle, "A fast software implementation for arithmetic operations in $GF(2^n)$," *Asiacrypt'96*, vol. 1163, pp. 65–76, 1996.