An Computation-Efficient Generalized Group-Oriented Cryptosystem

Ting-Yi CHANG

Graduate Institute of e-Learning National Changhua University of Education
No. 1, Jin-De Road, Changhua City, Taiwan, ROC
e-mail: tychang@cc.ncue.edu.tw

Received: February 2008; accepted: April 2010

Abstract. A *Group-Oriented Cryptosystem* (GOC) allows a sender to encrypt a message sent to a group of users so only the specified sets of users in that group can cooperatively decrypt the message. Recently, Li *et al.* pointed out unauthorized sets in the receiving group can recover the encrypted messages in Yang *et al.*'s GOC; and they further repaired this security flaw. However, the improved GOC contains inexact security analysis. Further, conversion of the scheme into a threshold cryptosystem results in inefficiency. This study enhances Li *et al.*'s GOC, both in that it achieves the requirements of GOC but also that it can be efficiently converted into a threshold cryptosystem. Under the decisional Diffie–Hellman problem assumption, the proposed scheme is demonstrated to be provably secure against chosen plaintext attacks.

Keywords: group-oriented cryptosystem, threshold cryptosystem, access structure, provable security, decisional Diffie–Hellman problem.

1. Introduction

Rapid advances in computer technology and the growth of the Internet are changing the way we conduct our daily and business lives. Secrecy is an important issue with respect to sensitive data transferred over insecure public channels. In traditional public-key cryptosystems such as RSA (Rivest *et al.*, 1978) and ElGamal (1985), any sender can use a recipient's public key to encrypt messages, and only the recipient who has the corresponding secret key is allowed to recover the message. However, some applications require several users be able to cooperatively decrypt messages for distributing the power of decrypting. For example, a sender may want to send a confidential contract for business to a group so only the managers can cooperatively recover the message. Unfortunately, traditional public-key cryptosystems cannot satisfy the requirements of the above situation.

To satisfy the necessary requirements of our daily lives including business activities, Desmedt (1987) proposed the concept of *Group-Oriented Cryptosystems* (GOC). In GOCs, a sender first determines an access structure suitable to a receiving group of users, and then sends an encrypted message to the group so only the authorized subsets of users in the group can cooperatively recover the message. An authorized subset of the receiving group is denoted an access instance f, and access instances are collectively known as the access structure F, which can be represented by the *Disjunctive Normal Form* (DNF), that is, $F = f_1 + f_2 + \cdots + f_k$. Let $U_1, U_2, \ldots, U_n$ be all users in the receiving group, and the access instances $f_1 = U_1U_2U_3$, $f_2 = U_2U_3U_4$, $f_3 = U_1U_4$. The access structure can be represented as $F = f_1 + f_2 + f_3 = U_1U_2U_3 + U_2U_3U_4 + U_1U_4$. Only U_1, U_2, and U_3; or U_2, U_3, and U_4; or U_1 and U_4 can cooperatively recover the message.

Yang *et al.* (2003) proposed a GOC based on the ElGamal cryptosystem, which is more efficient than previously proposed GOCs (Chang and Lee, 1992, 1993; Lin and Chang, 1994; Tsai *et al.*, 1999) in terms of the sender's computational complexity and the ciphertext size. Simultaneously, they also presented the elliptic curve version (Koblitz *et al.*, 2000) of the proposed GOC which provides smaller key sizes, increased bandwidth savings, and faster implementations. Recently, Li *et al.* (2007) pointed out an unauthorized set of users can recover messages sent using Yang *et al.*'s GOC, and proposed an improved method to withstand such attacks.

Li *et al.*'s improved GOC maintained the efficiency of Yang *et al.*'s GOC without any computational efforts or ciphertext sizes. The sender encrypts the message for each access instance by only multiplying the users' public keys in that instance. However, Li *et al.* did not provide an exact security analysis for the improved GOC. Another special concept of group-oriented cryptosystems is the (t, n) threshold cryptosystem, which allows any t users in the group of n users to have the ability to cooperatively recover the message and is also applied in many kinds of digital signature (Gao *et al.*, 2009; Liu and Huang, 2010). However, Li *et al.*'s GOC should require huge computational efforts and additional ciphertexts needed for conversion into a threshold cryptosystem. Thus, it is impractical in real-world applications.

In this study, a computation-efficient *Generalized Group-Oriented Cryptosystem* (GGOC) based on Pedersen's threshold cryptosystem (Pedersen, 1991a) is presented which is more general than Li *et al.*'s GOC. Though the proposed GGOC is based on the threshold cryptosystem, it still satisfies the requirements of GOC and is more efficient than Li *et al.*'s GOC converted into a threshold cryptosystem. Further, a sender will have no idea who cooperatively recovers messages in the threshold version; the proposed scheme can easily insert supervisors into the access structure to prevent any users with the threshold value from unscrupulously recovering the message. Compared to previously proposed GOCs, with respect to being converted into a threshold cryptosystem, the proposed GGOC requires a number of modular exponentiations for the sender and the ciphertext size for the group to be fixed as constants.

We also provide a concrete analysis of the reduction from *Decisional Diffie–Hellman* (DDH) to the proposed scheme for proving its security. Under the DDH problem assumption, the proposed GGOC is demonstrated to be provably secure against chosen plaintext attacks. As we all know, in public-key cryptosystems, the adversary can obtain the ciphertext of any plaintext chosen by him or her. This is a basic security demand for public-key cryptosystems (Bellare *et al.*, 1998, 2004).

The rest of this paper is organized as follows. In Section 2, Li *et al.*'s GOC is briefly reviewed and then some comments on the scheme are presented. Section 3 presents the

GGOC and shows its accuracy. Section 4 defines the security notations for the GGOC, and analyzes its security. Then, we compare the computational complexity and the ciphertext size of our scheme with those of Li *et al.*'s scheme. Finally, conclusions are specified in Section 5.

2. Comments on Li *et al.*'s Group-Oriented Cryptosystem

To state our results clearly and precisely, we begin with a review of Li *et al.*'s GOC. Assume U is the sender and $U_1, U_2, \dots, U_n$ are all users in the receiving group. Let p be a large prime so $p = 2q + 1$, and a generator g with order for the subgroup $\mathcal{G}$, where $\mathcal{G}$ is a subgroup of quadratic residues in $\mathbb{Z}_p^*$. The scheme is comprised of three phases: (1) *Key Generation Phase*, (2) *Encryption Phase*, and (3) *Decryption Phase*.

Key Generation Phase. The key distribution center chooses a random number $x_i \in_R \mathbb{Z}_q^*$ as a secret key for user U_i; and $y_i = g^{x_i} \bmod p$ as the corresponding public key for $i = 1$ to n in the receiving group. Anyone can via U_i's identity id_i to get his/her public key y_i from the X.509 directory authentication service (ITU, 2005).

Encryption Phase. To encrypt the message M $(M < p)$ for the group, the sender U firstly determines the access structure $F = f_1 + f_2 + \cdots + f_k$ for M, and then performs the following steps.

Step 1. Choose a random number $r \in_R \mathbb{Z}_q^*$ and compute $B = g^r \bmod p$.

Step 2. Compute $C_j = M \oplus ((\prod_{U_i \in f_j} y_i)^r \bmod p)$ for $j = 1$ to k, where $\oplus$ denotes the bit-wise exclusive-or operation. Then send $\{F, B, C_1, C_2, \dots, C_k\}$ to the receiving group.

Decryption Phase. After receiving $\{F, B, C_1, C_2, \dots, C_k\}$, the users U_is for $i = j_1$ to j_v in the access instance f_j use their secret keys to cooperatively recover the message M as follows.

Step 1. Compute $T_i = b^{x_i} \bmod p$ for $i = j_1$ to j_v.

Step 2. Recover $M = C_j \oplus (\prod_{U_i \in f_j} T_i \bmod p)$.

The sender encrypts a message for each access instance by multiplying the users' public keys in Li *et al.*'s GOC. There is no security proof to demonstrate what the cryptographic assumptions in Li *et al.*'s GOC are. Unfortunately, such a heuristic security analysis for evaluating the security results in several proposed schemes that were assumed to be secure but were later found to have been flawed (Choo *et al.*, 2005, 2006; Choo, 2008).

On the other hand, for practical applications using GOC, the receiving group is represented by the company, the access instances $f_1, f_2, \dots, f_k$ by the departments, and the users $U_1, U_2, \dots, U_n$ by the employees. Li *et al.*'s GOC allows senders to assign the employees by determining the access structure and to allow the users to cooperatively recover the message.

However, if the sender wants any t $(t \leqslant n)$ employees to collectively have the ability on behalf of the company or department to recover the message, the computational complexity of the sender will dramatically increase. Modifying the access structure F allows the GOC to be easily converted into the (t, n) threshold cryptosystem. That is, the access structure with the threshold value t can be represented as $F = U_1U_2 \ldots U_t + U_1U_2 \ldots U_{t-1}U_{t+1} + \cdots + U_{n-t+1}U_{n-t+2} \ldots U_n$. The number of access instances is $k = n!/t!(n - t)!$. Since the times of encryption a message M is according to the number of access instances k, Li *et al.*'s GOC requires enormous computational efforts and additional ciphertexts needed for conversion into a threshold cryptosystem. Indeed, the sender only uses the group's public key to encrypt the message M in Step 2 of the encryption phase in the threshold cryptosystems, only requiring 1 modular exponentiation. Compared with proper threshold cryptosystems, the sender's computation requires $(n!/t!(n-t)!)$ modular exponentiations in the original GOC, which are extremely inefficient. Of course, the ciphertext size is larger than that in a threshold cryptosystem.

Further, in the original GOC, the sender can utilize the identities to produce the access structure F and thus be aware of who is able to obtain the message. However, when the GOC is converted into the threshold cryptosystem, the sender has no knowledge of who will obtain the message. In order to prevent any t employees from unscrupulously recovering the confidential message, sometimes the threshold cryptosystem requires one or more supervisors to help in the process of decrypting. The access structure with the threshold value t and a supervisor S can be represented as $F = U_1U_2 \ldots U_tS + U_1U_2 \ldots U_{t-1}U_{t+1}S + \cdots + U_{n-t+1}U_{n-t+2} \ldots U_nS$. For the same reason, when the GOC is converted into the threshold cryptosystem with one or more supervisors, the process remains inefficient.

To solve the above problems, a computation-efficient generalized group-oriented cryptosystem is proposed in the next section.

3. The Proposed Generalized Group-Oriented Cryptosystem

The parameters p, q, and g are the same as those in Li *et al.*'s scheme. Pedersen's distributed key generation scheme (Pedersen, 1991a) based on verifiable secret sharing (Chang *et al.*, 2005; Pedersen, 1991b) is performed in the key generation phase. Details of three phases are stated as follows.

Key Generation Phase. Let $S_1, S_2, \ldots, S_l$ denote l supervisors, $x_{si} \in_R \mathbb{Z}_q^*$ the secret key, and $y_{si} = g^{x_{si}} \bmod p$ the corresponding public key. Each U_i for $i = 1$ to n in the receiving group performs the following steps:

Step 1. Choose a random number $d_i \in_R \mathbb{Z}_q^*$.

Step 2. Choose a random $(t - 1)$th degree polynomial $P_i(z)$ over $\mathbb{Z}_q^*$ such that $P_i(z) = p_{i,0} + p_{i,1}z + p_{i,2}z^2 \ldots p_{i,t-1}z^{t-1}$, where $\forall j$ $p_{i,j} \in \mathbb{Z}_q^*$ and $P_i(0) = p_{i,0} = d_i$. Then, send $P_i(id_j)$ to U_j via a secret channel and broadcast the check values $g^{p_{i,l}} \bmod p$ for $l = 0$ to $t - 1$.

After receiving $P_i(id_j)$ from U_i, each U_j checks its validity by the equation

$$g^{P_i(id_j)} \overset{?}{=} \prod_{l=0}^{t-1} g^{p_{i,l} \cdot id_j^l} \bmod p. \tag{1}$$

If Eq. (1) holds, proceed to the next step; else, $P_i(id_j)$ is requested to be sent again.

Step 3. Compute his/her secret key

$$x_i = \sum_{j=1}^{n} P_j(id_i) \bmod p, \tag{2}$$

the corresponding public key $y_i = g^{x_i} \bmod p$, and

$$y = \prod_{j=1}^{n} g^{p_{j,0}} \bmod p \tag{3}$$

is the public key for the group.

Encryption Phase. The sender U wants to send the message M to the group-oriented access structure F, the threshold value t of access structure F_t, or the threshold value t of access structure with supervisors F_t^s; and then performs the following steps in the individual cases.

Case I: For the access structure $F = f_1 + f_2 + \cdots + f_k$

Step 1. Choose a random number $r \in_R \mathbb{Z}_q^*$ and compute $B = g^r \bmod p$.

Step 2. Compute $C_j = M \oplus ((\prod_{U_i \in f_j} y_i)^r \bmod p)$ for $j = 1$ to k, and then send $\{F, B, C_1, C_2, \ldots, C_k\}$ to the group.

Case II: For the access structure $F_t = f_t$, where f_t denotes a threshold instance

Step 1. Choose a random number $r \in_R \mathbb{Z}_q^*$ and compute $B = g^r \bmod p$.

Step 2. Compute $C = M \oplus (y^r \bmod p)$ and send $\{F_t, B, C\}$ to the group.

Case III: For the access structure $F_t^s = f_t S_1 S_2 \ldots S_l$ with l supervisors

Step 1. Choose a random number $r \in_R \mathbb{Z}_q^*$ and compute

$$B = g^r \bmod p. \tag{4}$$

Step 2. Compute

$$C = M \oplus \left(\left(y \cdot \prod_{i=1}^{l} y_{si}\right)^r \bmod p\right), \tag{5}$$

and then send $\{F, B, C\}$ to the group.

Decryption Phase. According to the access structures F, F_t, or F_t^s, the users U_is for $i = j_1$ to j_v in the access instance f_j, or any t users in the group (with/without supervisors) can cooperatively recover the message M in the following three cases.

Case I: For the access structure F

Step 1. Compute $T_i = B^{x_i} \bmod p$ for $i = j_1$ to j_v.

Step 2. Recover

$$M = C_j \oplus \left(\prod_{U_i \in f_j} T_i \bmod p \right). \tag{6}$$

Case II: For the access structure F_t

Without loss of generality, assume that $U_1, U_2, \ldots, U_t$ want to recover the message M.

Step 1. Compute $T_i = B^{x_i \prod_{j=1, j \neq i}^{n} \frac{id_j}{id_j - id_i}} \bmod p$, for $i = 1$ to t.

Step 2. Recover

$$M = C \oplus \left(\prod_{i=1}^{t} T_i \bmod p \right). \tag{7}$$

Case III: For the access structure F_t^s

Each U_i for $i = 1$ to t and S_j for $j = 1$ to l performs the following steps, respectively.

Step 1. Compute

$$T_i = B^{x_i \prod_{j=1, j \neq i}^{n} \frac{id_j}{id_j - id_i}} \bmod p, \quad \text{for } i = 1 \text{ to } t. \tag{8}$$

Step 2. Compute

$$T_{sj} = B^{x_{sj}} \bmod p, \quad \text{for } j = 1 \text{ to } l. \tag{9}$$

Step 3. Recover

$$M = C \oplus \left(\prod_{i=1}^{t} T_i \cdot \prod_{j=1}^{l} T_{sj} \bmod p \right). \tag{10}$$

Pedersen's distributed key generation scheme is employed during the key generation phase, and thus is unrelated to the functions of Li *et al.*'s GOC. Any user U_j in the group can verify that $P_i(id_j)$ is distributed by U_i in Eq. (1) (Pedersen, 1991b). Obviously, Case I is Li *et al.*'s GOC and Case II is Pedersen's threshold cryptosystem. The sender encrypts the message by multiplying the group's public key and the supervisors' public keys in Case III. One can see that Case III includes Case I and Case II. The correctness of Eq. (6) and Eq. (7) can be shown by Eq. (10). Here, the correctness of Eq. (10) is shown as follows. From Eq. (5), we have

$$\begin{aligned} C &= M \oplus \left(\left(y \cdot \prod_{i=1}^{l} y_{si}\right)^{r} \bmod p\right) \\ &= M \oplus \left(\left(g^{\sum_{j=1}^{n} p_{j,0}} \cdot g^{\sum_{i=1}^{l} x_{si}}\right)^{r} \bmod p\right) \quad \text{by Eq. (3)} \\ &= M \oplus \left(B^{\sum_{j=1}^{n} p_{j,0}} \cdot B^{\sum_{i=1}^{l} x_{si}} \bmod p\right) \quad \text{by Eq. (4).} \end{aligned}$$

The above equation can be further rewritten as

$$\begin{aligned} M &= C \oplus \left(B^{\sum_{j=1}^{n} p_{j,0}} \cdot B^{\sum_{i=1}^{l} x_{si}} \bmod p\right) \\ &= C \oplus \left(B^{\sum_{i=1}^{t} x_i \prod_{j=1, j\neq i}^{n} \frac{id_j}{id_j - id_i}} \cdot B^{\sum_{i=1}^{l} x_{si}} \bmod p\right) \\ &\quad \text{by the Lagrange formula} \\ &= C \oplus \left(\prod_{i=1}^{t} T_i \cdot \prod_{j=1}^{l} T_{sj} \bmod p\right) \end{aligned}$$

which lead to Eq. (10).

4. Security Analysis and Performance Evaluation

In this section, several security notations used in the generalized group-oriented cryptosystem are defined, and then the security and performance of the proposed scheme are analyzed. Since Case III in the proposed scheme includes Case I and Case II, the following narrations are aimed at Case III.

4.1. *Security Notations*

In order to prove the GGOC scheme is secure against chosen-plaintext attacks under the DDH problem, a variation of the DDH problem is introduced. Since the sender encrypts a message by multiplying both the group's public key and the supervisors' public keys, this problem is called the *Multiply Decisional Diffie–Hellman* (MDDH) problem. Subsequently, we shall prove a polynomial-time reduction from the DDH problem to the MDDH problem.

DEFINITION 1 [Decisional Diffie–Hellman Problem]. *There are several domain parameters, primes p and q such that $q|p-1$, and a generator g with order for the subgroup $\mathcal{G}$, where $\mathcal{G}$ is a subgroup of quadratic residues in $\mathbb{Z}_p^*$. Consider the following two distributions with $\alpha, \beta \in_R \mathbb{Z}_q^*$ and $Z \in_R \mathcal{G}$:*

- *the distribution* $\mathbf{R}$ *of* $(g^{\alpha}, g^{\beta}, Z) \in \mathcal{G}^3$*;*
- *the distribution* $\mathbf{D}$ *of* $(g^{\alpha}, g^{\beta}, g^{\alpha\beta}) \in \mathcal{G}^3$.

The DDH problem is hard if there is no polynomial-time algorithm $\mathcal{A}$ that satisfies

$$|\Pr[\mathcal{A}(\mathbf{D}) = \mathit{true}] - \Pr[\mathcal{A}(\mathbf{R}) = \mathit{true}]| > \frac{1}{P(|q|)},$$

for any polynomial P, where the probability concerns the random choices α, β and Z.

ASSUMPTION 1 [Multiply Decisional Diffie–Hellman Problem]. *There are several domain parameters p, q, g, and $\mathcal{G}$ identical to those in Definition 1. Consider the following two distributions with $x_{si}s \in_R \mathbb{Z}_q^*$ for $i = 1$ to l, $x, r \in_R \mathbb{Z}_q^*$, and $N \in_R \mathcal{G}$:*

- *the distribution $\mathbf{R}'$ of $(g^{x_{s1}}, g^{x_{s2}}, \ldots, g^{x_{sl}}, g^x, g^r, N) \in \mathcal{G}^{l+3}$;*
- *the distribution $\mathbf{D}'$ of $(g^{x_{s1}}, g^{x_{s2}}, \ldots, g^{x_{sl}}, g^x, g^r, g^{(x_{s1}+x_{s2}+\cdots+x_{sl}+x)r}) \in \mathcal{G}^{l+3}$.*

The MDDH problem is hard if there is no polynomial-time algorithm $\mathcal{A}$ that satisfies

$$|\Pr[\mathcal{A}(\mathbf{D}') = \mathit{true}] - \Pr[\mathcal{A}(\mathbf{R}') = \mathit{true}]| > \frac{1}{P(|q|)},$$

for any polynomial P, where the probability concerns the random choices $x_{si}s, x, r$, and N.

The security of message confidentiality is defined as indistinguishability under chosen plaintext attacks in the following security model (Bellare *et al.*, 1998). In this security model, the adversary attempts to compromise a target ciphertext of the designated recipients (the group and supervisors).

DEFINITION 2 [Indistinguishability under Chosen Plaintext Attacks]. *A generalized group-oriented cryptosystem is semantically secure against chosen plaintext attacks if there exists no polynomial-time adversary $\mathcal{A}$ with a non-negligible advantage in the following game.*

SETUP*: First, the GGOC's key generation algorithm is run by a challenger $\mathcal{C}$ with a security parameter as input. $\mathcal{C}$ gives the system parameters, the group's public key y, and the supervisors' public keys $y_{si}s$ for $i = 1$ to l to the adversary $\mathcal{A}$. Note that $\mathcal{A}$ has no knowledge of the corresponding secret keys.*

CHALLENGE*: The adversary $\mathcal{A}$ chooses two equal length messages, M_0, M_1, and then sends these to the challenger $\mathcal{C}$. $\mathcal{C}$ chooses a bit $b \in_R \{0, 1\}$ which is not in $\mathcal{A}$'s view, and encrypts M_b. The corresponding ciphertext (B^*, C^*) is given to $\mathcal{A}$ as a challenge.*

GUESS*: At the end of the game, the adversary $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$, which is supposed to be $\mathcal{A}$'s guess of the value b. $\mathcal{A}$ wins this game if $b' = b$ and the advantage of $\mathcal{A}$ is defined as* $\mathrm{Adv}(\mathcal{A}) = \Pr[b' = b] - \frac{1}{2}$.

4.2. *Security Analysis*

The security of the proposed GGOC is based on Pedersen's threshold cryptosystem and the MDDH problem. In the following theorem, we surmise that Pedersen's threshold cryptosystem is secure and prove that the proposed GGOC satisfies the security requirements in Definition 2.

Theorem 1. *The proposed generalized group-oriented cryptosystem is* (t, ϵ)*-secure against chosen plaintext attacks if there is no polynomial-time algorithm that solves the MDDH problem with* (t', ϵ')*.*

Proof. Proof by the reduction to contradiction: assume that there exists an (t, ϵ)-adversary $\mathcal{A}$ who can break the GGOC in the game concerning Definition 2, where t is the running time and ϵ is the advantage that $\mathcal{A}$ succeeds. We demonstrate how to construct an (t', ϵ')-algorithm $\mathcal{B}$ that solves the MDDH problem with running time t' and advantage ϵ'. Given the distribution either from $\mathbf{R}'$ or $\mathbf{D}'$ in Assumption 1, $\mathcal{B}$ simulates the joint distribution consisting of $\mathcal{A}$'s view in its attack on the proposed scheme and the hidden bit b. The details of the simulation are as follows.

SETUP: With the distribution either from $\mathbf{R}'$ or $\mathbf{D}'$, $\mathcal{B}$ starts to simulate $\mathcal{A}$'s challenger and outputs the system parameters p, q, g, the group's public key y as $(l + 1)$th element of the distribution; and the supervisors' public keys y_{si}s as ith element of the distribution for $i = 1$ to l to the adversary $\mathcal{A}$.

CHALLENGE: The adversary $\mathcal{A}$ chooses two equal length messages, M_0, M_1, and then sends these to the challenger $\mathcal{C}$. $\mathcal{C}$ chooses a bit $b \in_R \{0, 1\}$. $\mathcal{B}$ treats B^* as $(l + 2)$th element of the distribution, and uses the last element σ ($\sigma = N$ or $\sigma = g^{(x_{s1}+x_{s2}+\cdots+x_{sl}+x)r}$) of the distribution to compute $C^* = M_b \oplus \sigma$. The ciphertext (B^*, C^*) is sent to $\mathcal{A}$ as a challenge. If σ is derived from $\mathbf{D}'$, then (B^*, C^*) is indeed a ciphertext of M_b. If σ is derived from $\mathbf{R}'$, then C^* is a random element, and hence (B^*, C^*) is independent of b.

GUESS: At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

Analysis: If the input comes from $\mathbf{D}'$ ($\alpha = g^{(x_{s1}+x_{s2}+\cdots+x_{sl}+x)r}$), the simulation of $\mathcal{B}$ will be nearly perfect, and $\mathcal{A}$ will have a non-negligible advantage ϵ in guessing the hidden bit b. Hence, $\Pr[\mathcal{B}(\mathbf{D}') = \mathit{true}] = \Pr[b = b']$. If the input comes from $\mathbf{R}'$ ($\alpha = N$), then $\mathcal{A}$'s view is essentially independent of b, and hence the probability of it outputting $b = b'$ is at most $\frac{1}{2}$. Therefore, $\mathcal{B}$'s advantage $\mathrm{Adv}(\mathcal{B}) = \epsilon' \geqslant \Pr[b = b'] - \frac{1}{2} \geqslant \epsilon$. From the specification of $\mathcal{B}$, the running time t' counts can be in a polynomial-time.

Theorem 1 has shown that an adversary wanting to break the security of the proposed generalized group-oriented cryptosystem would face the MDDH problem. However, further evidence of the trustworthiness of the MDDH problem in Assumption 1 should be provided. The following lemma shows that the MDDH problem is at least as hard as the DDH problem.

Lemma 1. *The MDDH problem is at least as hard as the DDH problem.*

Proof. Proof by contradiction: assume that there exists an algorithm $\mathcal{A}$ that can efficiently distinguish the distributions $\mathbf{D}'$ and $\mathbf{R}'$, where $x, x_{si}\mathrm{s}, r \in_R \mathbb{Z}_q^*$, and $N \in_R \mathcal{G}$. We demonstrate that an algorithm $\mathcal{B}$ can be constructed by using the algorithm $\mathcal{A}$ to efficiently distinguish the distributions $\mathbf{D}$ and $\mathbf{R}$, where $\alpha, \beta \in_R \mathbb{Z}_q^*$, and $Z \in_R \mathcal{G}$. First, either $\mathbf{D}$ or $\mathbf{R}$ is the input of algorithm $\mathcal{B}$. Assume that the input of $\mathcal{B}$ is $\mathbf{E}$ and $(\mathbf{E})_i$ denotes ith element of the distribution $\mathbf{E}$. $\mathcal{B}$ chooses $x_{si} \in_R \mathbb{Z}_q^*$ for $i = 1$ to l and produces the following distribution for $\mathcal{A}$:

$$\left(g^{x_{s1}}, g^{x_{s2}}, \ldots, g^{x_{sl}}, (\mathbf{E})_1, (\mathbf{E})_2, (\mathbf{E})_3 \cdot \left(\prod_{i=1}^{l} (\mathbf{E})_2^{x_{si}}\right) \bmod p\right) \in \mathcal{G}^{l+3}$$

Analysis: If the input comes from $\mathbf{D}$, the last element would be:

$$\begin{aligned}(\mathbf{E})_3 \cdot \left(\prod_{i=1}^{l} (\mathbf{E})_2^{x_{si}}\right) &= g^{\alpha\beta} \cdot \left(\prod_{i=1}^{l} g^{\beta \cdot x_{si}}\right) \bmod p \\ &= g^{\alpha\beta} \cdot g^{\beta x_{s1} + \beta x_{s2} + \cdots + \beta x_{sl}} \bmod p \\ &= g^{(x_{s1} + x_{s2} + \cdots + x_{sl} + \alpha)\beta} \bmod p.\end{aligned}$$

$\mathcal{A}$ will have a non-negligible advantage in the probability of $\Pr[\mathcal{A}(\mathbf{D}') = \mathit{true}]$. Hence, $\Pr[\mathcal{B}(\mathbf{D}) = \mathit{true}] = \Pr[\mathcal{A}(\mathbf{D}') = \mathit{true}]$. If the input comes from $\mathbf{R}$, the last element would be:

$$(\mathbf{E})_3 \cdot \left(\prod_{i=1}^{l} (\mathbf{E})_2^{x_{si}}\right) = Z \cdot \left(\prod_{i=1}^{l} g^{\beta \cdot x_{si}}\right) \bmod p.$$

Since Z is a random number in $\mathcal{G}$, the last element of the distribution from $\mathcal{A}$'s view is a random element; in other words, $\Pr[\mathcal{B}(\mathbf{R}) = \mathit{true}] = \Pr[\mathcal{A}(\mathbf{R}') = \mathit{true}]$ is negligible. Therefore, $\mathcal{B}$ can use $\mathcal{A}$ to efficiently distinguish the distributions $\mathbf{R}$ and $\mathbf{D}$, which is a contradiction of Assumption 1. Obviously, the reduction is in a polynomial-time.

In summary, the proposed GGOC is demonstrated to be secure against chosen plaintext attacks under the MDDH problem assumption in Theorem 1, and there is a polynomial-time reduction from the DDH problem to the MDDH problem as stated in Lemma 1. Any adversary wanting to break the security of proposed GGOC under chosen plaintext attacks will face the DDH problem.

4.3. *Performance Evaluation*

The proposed GGOC is efficient in that the scheme is converted into a threshold cryptosystem. For the access structure F in Case I, our GGOC is equal to Li *et al.*'s GOC. In this section, we compare the sender's computational complexity and the ciphertext size of our GGOC with those of Li *et al.*'s GOC in cases where schemes are extended to a (t, n) threshold cryptosystem with l supervisors. To facilitate the performance evaluations, we first define the following notations:

Table 1

Performance evaluations of the proposed GGOC and Li *et al.*'s GOC

	Sender's computational complexity	Ciphertext size
Our GGOC	$2 \times T_{\mathrm{EXP}} + (1+l) \times T_{\mathrm{MUL}}$ $1 \times T_{\mathrm{XOR}}$	$2\times$1024-bit
Li *et al.*'s GOC	$(1+\frac{n!}{t!(n-t)!}) \times T_{\mathrm{EXP}} + (t+l)\times$ $T_{\mathrm{MUL}} + (\frac{n!}{t!(n-t)!}) \times T_{\mathrm{XOR}}$	$(1+\frac{n!}{t!(n-t)!})\times$1024-bit

T_{EXP}: the time for computing a modular exponentiation computation,
T_{MUL}: the time for computing a modular multiplication computation,
T_{XOR}: the time for computing a bit-wise exclusive-or operation.

According to Table 1, it is obvious that the sender's computations in our scheme are fewer than in Li *et al.*'s scheme. Since T_{EXP} is much larger than T_{MUL} and T_{XOR}, the following descriptions only compare the number of T_{EXP}. The numbers of T_{EXP} in the sender's computations of Li *et al.*'s scheme increase by $\frac{n!}{t!(n-t)!}$ whereas our scheme is fixed by a constant 2. The ciphertexts of our scheme are (B, C) and Li *et al.*'s scheme are $(B, C_1, C_2, \ldots, C_{\frac{n!}{t!(n-t)!}})$. Assume that the modulus p is around 1024-bit in both schemes. The ciphertext sizes of our scheme and Li *et al.*'s scheme are $2\times$1024-bit and $(1+\frac{n!}{t!(n-t)!})\times$1024-bit, respectively. Thus it can be seen that the communication overhead for transferring the ciphertexts in our scheme is less than that in Li *et al.*'s scheme.

5. Conclusions

In this paper, we proposed a generalized group-oriented cryptosystem, which is more generalized than Li *et al.*'s scheme. The proposed GGOC allows senders to send the encrypted message to a group of n users, and only the specific recipients, or any t recipients (with or without supervisors) in the group can cooperatively recover the message according to the sender's determined access structures. In addition, the sender's computational complexity and the ciphertext size produce superior results to those of Li *et al.*'s scheme, as they are a threshold cryptosystem. Thus, our scheme is suitable for organizational operations in real-world applications.

Under the decisional Diffie–Hellman problem assumption, this study has demonstrated the proposed GGOC is provably secure against chosen plaintext attacks. It also implies Li *et al.*'s GOC is secure against chosen plaintext attacks under the same assumption.

Acknowledgements. The authors wish to thank many anonymous referees for their suggestions to improve this paper. Part of this research was supported by the National Science Council, Taiwan, ROC, under contract No. NSC97-2221-E-018-011 and NSC98-2622-E-018-004-CC3.

References

Bellare, M., Palacio, A. (2004). Towards plaintext-aware public-key encryption without random oracles. In: *Advances in Cryptology – ASIACRYPT'04, Lecture Notes in Computer Science*, Vol. 3329, pp. 48–62.

Bellare, M., Desai, A., Pointcheval, D., Rogaway, P. (1998). Relations among notations of security for public-key encryption schemes. In: *Advances in Cryptology – CRYPTO'98, Lecture Notes in Computer Science*, Vol. 1462, pp. 26–45.

Chang, C.C., Lee, H.C. (1992). A solution to generalized group oriented cryptography. In: *IFIP/Sec'92-Singapore Day2/Track2-Cryptography*, pp. 289–299.

Chang, C.C., Lee, H.C. (1993). A new generalized group-oriented cryptoscheme without trusted centers. *IEEE Journal on Selected Area in Communications*, 11(5), 725–729.

Chang, T.Y., Hwang, M.S., Yang, W.P. (2005). An improvement on the Lin–Wu (t, n) threshold verifiable multi-secret sharing scheme. *Applied Mathematics and Computation*, 163(1), 169–178.

Choo, K.K.R. (2008). *Secure Key Establishment*. Springer, Berlin.

Choo, K.K.R., Boyd, C., Hitchcock, Y. (2005). Errors in computational complexity proofs for protocols. In: *Advances in Cryptology – Asiacrypt 2005, Lecture Notes in Computer Science*, Vol. 3788, pp. 624–643.

Choo, K.K.R., Boyd, C., Hitchcock, Y. (2006). The importance of proofs of security for key establishment protocols formal analysis of Jan–Chen, Yang–Shen–Shieh, Kim–Huh–Hwang–Lee, Lin–Sun–Hwang, and Yeh–Sun protocols. *Computer Communications*, 29(15), 2788–2797.

Desmedt, Y. (1987). Society and group oriented cryptography: a new concept. In: *Advances in Cryptology – CRYPTO'87, Lecture Notes in Computer Science*, Vol. 293, pp. 120–127.

ElGamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31, 469–472.

Gao, W., Wang, G., Wang, X., Yang, Z. (2009). One-round ID-based threshold signature scheme from bilinear pairings. *Informatica*, 20(4), 461–476.

ITU-X Recommendation X.509 (2005). Information technology – open systems interconnection – the directory: Authentication framework.

Koblitz, N., Menezes, A., Vanstone, S.A. (2000). The state of elliptic curve cryptography. *Designs, Codes and Cryptography*, 9(2/3), 173–193.

Li, C.M., Hwang, T., Lee, N.Y. (2007). Security flaw in simple generalized group-oriented cryptosystems using ElGamal cryptosystem. *Informatica*, 18(1), 61–66.

Lin, C.H., Chang, C.C. (1994). Method for constructing a group-oriented cipher system. *Computer Communications*, 17(11), 805–808.

Liu, J., Huang, S. (2010). Identity-based threshold proxy signature from bilinear pairings. *Informatica*, 21(1), 41–56.

Pedersen, T.P. (1991a). A threshold cryptosystem without a trusted party. In: *Advances in Cryptology – CRYPTO'91, Lecture Notes in Computer Science*, Vol. 547, pp. 522–526.

Pedersen, T.P. (1991b). Non-interactive and information-theoretic verifiable secret sharing. In: *Advances in Cryptology – CRYPTO'91, Lecture Notes in Computer Science*, Vol. 576, pp. 129–140.

Rivest, R.L., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21, 120–126.

Tsai, J.J., Hwang, T., Wang, C.H. (1999). New generalized group-oriented cryptosystem based on Diffie–Hellman scheme. *Computer Communications*, 22(8), 727–729.

Yang, C.C., Chang, T.Y., Li, J.W., Hwang, M.S. (2003). Simple generalized group-oriented cryptosystems using ElGamal cryptosystem. *Informatics*, 14(1), 111–120.

T.-Y. Chang received the BS in information management from Chaoyang University of Technology (CYUT), Taiwan, Republic of China, in 2001; and the MS degree in information and communication engineering from CYUT in 2003 and the PhD degree in computer science from National Chiao Tung University in 2006. He is currently an associate professor in the Graduate Institute of e-Learning, National Changhua University of Education, Taiwan. His research interests include information security, cryptography, mobile communications, and e-learning.

Efektyvi apibendrintoji grupinė orientuotoji kriptosistema

Ting-Yi CHANG

Grupinė orientuotoji kriptosistema (GOK) suteikia siuntėjui galimybę užšifruoti pranešimą siunčiamą vartotojų grupei taip, kad tik tam tikri grupės vartotojai gali bendradarbiaudami šį pranešimą iššifruoti. Li ir kt. pastebėjo, kad kai kurie neautorizuoti grupės vartotojai gali iššifruoti Yang ir kt. GOK pranešimus, todėl jie šį metodą patobulino. Tačiau nėra atlikta šios schemos saugos analizė, o jos transformavimas į slenkstinę kriptosistemą yra neefektyvus.

Straipsnyje pateikta patobulinta Li ir kt. GOK ne tik tenkina GOK reikalavimus, bet ir gali būti transformuota į slenkstinę kriptosistemą. Jei tenkinamos Diffie–Hellman'o uždavinio sąlygos, tuomet pasiūlytas šifravimo algoritmas yra saugus atakoms, kai įsilaužėlis gali pasirinkti neužšifruotus tekstus.