GO-CP-ABE: group-oriented ciphertext-policy attribute-based encryption

Yang Li
Fujian Provincial Key Laboratory of Network Security and Cryptology,
School of Mathematics and Computer Science,
Fujian Normal University,
Fuzhou, China
Email: 1175168008@qq.com

Xiaoling Tao*
Key Laboratory of Cognitive Radio and Information Processing,
University of Electronic Technology,
Guilin 541004, China
and
Guangxi Colleges and Universities Key Laboratory of
Cloud Computing and Complex Systems,
Guilin University of Electronic Technology,
Guilin 541004, China
Email: txl@guet.edu.cn
*Corresponding author

Wei Wu
Fujian Provincial Key Laboratory of Network Security and Cryptology,
School of Mathematics and Computer Science,
Fujian Normal University,
Fuzhou, China
Email: weiwu81@gmail.com

Joseph K. Liu
Faculty of Information Technology,
Monash University, Australia
Email: ksliu9@gmail.com

Abstract: We introduce a variant of ciphertext-policy attribute-based encryption called group-oriented ciphertext-policy attribute-based encryption (GO-CP-ABE). In the new notion, message is encrypted under an access structure over attributes and users are described by their attributes. In addition, users are divided into different groups and each user belongs to only one group. Users within the same group can merge their attributes to decrypt successfully, if the union of their attributes satisfies the access structure embedded in the ciphertext. But users from different groups cannot realise this cooperative decryption. We define the security model of GO-CP-ABE and present an efficient design by revising an existing CP-ABE scheme.

Keywords: attribute-based encryption; group-oriented; access structure; cooperative decryption.


Biographical notes: Yang Li received his Bachelor’s degree from the School of Institute of Electrical and Information Engineering, Hubei University of Automotive Technology, China in 2013. He is currently a graduate student at the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. His research interests include cryptography and information security.

Xiaoling Tao received her MS in Computer Application Technology from the Guilin University of Electronic Technology, Guilin, China in 2008 and became a faculty member of this university in 2000. She is currently an Associate Professor at the school of information and communication,
1 Introduction

Traditional public-key encryption is a powerful tool to protect data confidentiality during storage and transmission. This encryption mechanism can realise secure data sharing only with a target user, or say in another way, it allows one to encrypt data for a specific recipient, but cannot efficiently handle more expressive types of encrypted access control. Therefore, Sahai and Waters (2005) presented a new vision for encryption where the data provider can express how he wants to share data. With this new notion, the data provider can embed a predicate \( f(\cdot) \) during encryption to describe how he wants to share the data and the user’s secret key is associated with their credentials \( X \). Anyone can decrypt successfully if and only if his/her credential satisfies the predicate. Sahai and Waters (2005) first presented the notion of attribute-based encryption (ABE), in which the user’s credential is represented by a set of attributes and the predicate is represented by a formula over attributes. But their scheme does not support expressive predicate and only supports threshold access control. In order to have more expressive predicate, in the later work, Goyal et al. [Goyal et al. (2006); Ostrovsky et al. (2007)] defined the notions of ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE). In CP-ABE, attributes are used to describe user’s secret keys and formulas over attributes are attached to ciphertext. On the contrary, in KP-ABE, attributes are used to annotate ciphertext and formulas over attributes are ascribed to user’s secret keys. In addition to the traditional confidentiality requirement, all existing KP-ABE and CP-ABE schemes are designed to be collusion-resistant, which is believed as an critical property for building attribute-based cryptographic access control systems. The property of collusion-resistance disables users to decrypt ciphertexts by collusion. Namely, users can pool their attributes, but they cannot decrypt the ciphertexts if none of them is able to satisfy the encryption policy individually. However, this property is not always preferable in practical applications. As an example, ABE can be used to realise fine-grained secure sharing of personal health record (PHR) in cloud computing. In a nutshell, one’s health records are encrypted by CP-ABE and the encrypted records are uploaded to the cloud. Anyone, such as doctors, friends and insurance company staffs, is able to decrypt and retrieve the records if his/her attributes satisfy the access policy. As all health-related systems, ABE-based PHR systems are required to support ‘break-glass’ operations and ensure that encrypted health records can be decrypted in emergency situations. However, a special case exists in the ABE-based PHR systems. For example, a patient’s PHR is encrypted under an access structure (‘dermatologist’ and ‘gastroenterologist’), but there is no doctor who is a specialist in both dermatosis and cardiopathy. In this case, two doctors need to decrypt the record collaboratively. As a result, a normal ABE scheme cannot be used in this scenario. Motivated by this observation, the notion of group collaboration was proposed for the first time in group-oriented attribute-based encryption (GO-ABE), and this is also the motivation of our paper. Then, in Li et al. (2015), the concept of group collaboration was used in signature. Compared with normal ABE, in GO-ABE, an added feature is that users within the same group can decrypt ciphertexts by merging their attributes (and private keys). The decryption would be successful if the union of their attributes satisfies the access policy, but users from different groups cannot decrypt ciphertexts by collusion. The scheme proposed in Li et al. (2014) only supports threshold formula, rather than an expressive one. Based on this observation, we aim to propose a design of group-oriented ciphertext-policy attribute-based encryption (GO-CP-ABE) supporting more expressive formulas.
1.1 Our contributions

In this paper, we propose the concept of GO-CP-ABE with more expressive formulas. Compared with the GO-ABE with threshold structure, the scheme proposed in this paper supports monotone access tree structure. In our scheme, users within the same group can pool their decryption keys to decrypt a ciphertext encrypted under an access structure which none of them may be able to satisfy the access structure individually, but users from different groups cannot make it. We provide a concrete construction of our concept based on the CP-ABE scheme in Waters (2011). Our decryption algorithm requires less number of bilinear pairing operations than the original algorithm presented in Waters (2011). Besides, we prove the security of our scheme. Specifically, the security of our scheme can be reduced to the decisional $q$-parallel BDHE problem.

1.2 Organisation

The remainder of our paper is organised as follows. In Section 2, we present related work. Section 3 is devoted to the preliminaries required by this paper. In Section 4, we describe the definition of GO-CP-ABE. In Section 5, we give a detailed description of our scheme. We then prove the security of our scheme in Section 6. In section 7, we analyse the efficiency of our scheme in terms of time complexity. Section 8 concludes this paper.

2 Related work

Sahai and Waters (2005) first put forward the concept of ABE and also introduced a scheme called fuzzy identity-based encryption (FIBE). The scheme of FIBE builds upon several ideas from identity-based encryption (IBE) (Boneh and Franklin, 2001; Cocks, 2001; Shamir, 1984). ABE is a new encryption method, compared with the encryption in traditional public key cryptography, such as IBE (Luo, 2015; Wang et al., 2015). In ABE, user’s private keys and ciphertexts are associated with a set of attributes and a policy over attributes. Anyone is able to decrypt a ciphertext if there is a ‘match’ between his/her private key and the ciphertext. The original ABE system introduced by Sahai and Waters (2005) is a threshold ABE system in which the ciphertext is labelled with a set of attributes $\alpha$ and the user’s private key is not only associated with a threshold parameter $\kappa$ but associated with another set of attributes $\beta$. A user can decrypt a ciphertext if and only if at least $\kappa$ attributes overlap between attribute sets $\alpha$ and $\beta$. One of the primary motivations for this was to design an error-tolerant or FIBE scheme that could use biometric identities. The primary drawback of the threshold ABE introduced by Sahai and Waters (2005) is that the threshold semantics is not very expressive and therefore is not suitable for designing more general systems.

Goyal et al. (2006) clarified the concept of ABE into KP-ABE and CP-ABE. In recent years, a great number of ABE schemes have been proposed.

- KP-ABE (such as Goyal et al., 2006; Ostrovsky et al., 2007; Attrapadung et al., 2011; Hohenberger and Waters, 2013). Goyal et al. (2006) introduced the idea of a more general KP-ABE. In KP-ABE, the private key is associated with a policy and the ciphertext is associated with a set of attributes. The construction of Goyal et al. (2006) can be viewed as an extension of the Sahai and Waters (2005) technique where instead of embedding a Shamir (1979) secret sharing scheme in the private key, the authority embeds a more general secret sharing scheme for the monotone access tree. To enable more flexible access control policy, Ostrovsky et al. (2007) presented a KP-ABE scheme that supports non-monotone access structure. Recently, KP-ABE has been widely used in cloud storage (Yu et al., 2010a; Lee et al., 2013; Li et al., 2013) to protect the data security. In 2014, Hohenberger and Waters proposed a design of online/offline attribute-based encryption.

- CP-ABE (such as Bethencourt et al., 2007; Lewko et al., 2010; Yu et al., 2010b). In contrast, in CP-ABE, attributes are used to label the user’s private key and the policy over attributes is attached to ciphertext. The first CP-ABE scheme that allows any monotone access structures with the security proof in the generic bilinear group model was proposed by Bethencourt et al. in 2007. In 2011, Lai et al. proposed a scheme of hiding the access structure associated with ciphertexts. Now, CP-ABE has a wide range of applications, such as social networking sites and electronic medical systems (Qin et al., 2015; Yan et al., 2016).

In order to further enrich the application of ABE, a new variant of ABE has been proposed. It is called GO-ABE (Li et al., 2014). Compared with normal ABE, an added feature is that users within the same group can decrypt by merging their attributes (and private keys). The decryption would be successful if the union of their attributes satisfies the access policy. But users from different groups cannot make it. However, the scheme proposed in Li et al. (2014) only supports a threshold formula, rather than an expressive one.

3 Preliminaries

In this part, we first briefly review the bilinear pairings. Then, we present a definition of monotone access structure. Finally, we give some background information on linear secret sharing scheme and state the complexity assumption we use for our proof of security.

3.1 Bilinear pairings

We briefly present a few facts about groups with efficiently computable bilinear pairings. Let $G_1$ and $G_2$ be two multiplicative cycle groups of prime order $p$. $g$ is a generator of $G_1$, $e$ is a bilinear pairing, $e: G_1 \times G_1 \rightarrow G_2$, has the following properties:
The advantage of an access structure is a collection \( A \subseteq 2^{\{1, \ldots, n\}} \) monotone if for all subsets \( B \subseteq C \) then \( B \subseteq A \). An access structure is a collection \( A \) of non-empty subsets of \( \{P_1, P_2, \ldots, P_n\} \); i.e., \( A \subseteq 2^{\{1, \ldots, n\}} \setminus \{\emptyset\} \). The sets in \( A \) are called the authorised sets, and the sets not in \( A \) are called the unauthorised sets.

In our scheme, we restrict our attention to monotone access structure.

### 3.2 Access structure

**Definition 3.1:** [Access structure (Beimel, 1996)] Let \( \{P_1, P_2, \ldots, P_n\} \) be a set of parties. A collection \( A \subseteq 2^{\{1, \ldots, n\}} \) is called the authorised sets, and the sets not in \( A \) are called the unauthorised sets.

It is shown in Beimel (1996) that every linear secret sharing scheme according to the above definition also enjoys the linear reconstruction property, defined as follows: Suppose that \( \Pi \) is a linear secret sharing scheme for the access structure \( \Upsilon \). Let \( A \in \Upsilon \) be any authorised set, and let \( l \subseteq \{1, 2, \ldots, l\} \) be defined as \( I = \{i \mid \rho(i) \in A\} \). Then, there exist constants \( \{\alpha_i \in \mathbb{Z}_p\}_{i \in l} \) such that, if \( \ell_i = (M\bar{v})_i \) are valid shares of secret \( s \) according to \( \Pi \), then \( \Sigma_{i \in l} \alpha_i \ell_i = s \).

Furthermore, it has been proved in Beimel (1996) that these constants \( \{\alpha_i\} \) can be found in polynomial-time in the size of share-generating matrix \( M \).

### 3.4 Complexity assumption

**Definition 3.3:** (Decisional parallel bilinear Diffie-Hellman exponent (BDHE) assumption) Let \( a, s \in \mathbb{Z}_p \) be chosen at random and \( g \) be a generator of group \( G_1 \) of prime order \( p \). The decision \( q \)-BDHE introduced by Boneh et al. (2005) that no probabilistic polynomial-time adversary \( A \) can distinguish the \( e(g, g)^{sv_i} \) from a random element \( R \in G_2 \) with more than a non-negligible advantage when given \( \bar{v} = (g, g^{v_1}, g^{v_2}, \ldots, g^{v_N}) \). The advantage of an algorithm \( B \) in solving decisional \( q \)-BDHE is defined as follows:

\[
\Pr\left[ B(\bar{v}, e(g, g)^{sv_i}) = 0 \right] - \Pr\left[ B(\bar{v}, R) = 0 \right] \geq \epsilon
\]

### 4 Definitions of GO-CP-ABE

In GO-CP-ABE, we divide users into different groups. There is no user belonging to two or more groups. Like CP-ABE the ciphertext is encrypted under an access structure and user’s private key is associated with attributes. Only users from the same group can merge their private keys to decrypt ciphertexts, but users from different groups cannot make it. In other words, users from the same group are able to cooperate with each other to decrypt ciphertexts encrypted under an access structure satisfied by the union of their attributes. But users from different groups cannot collude to decrypt ciphertexts. A GO-CP-ABE scheme consists of four algorithms and is described as follows:

1. **Setup(\text{U})**
   - This is a randomised algorithm that takes a security parameter and an attribute universe as input. It outputs the public parameters \( \text{PK} \) and a master key \( \text{MK} \).
   
   \( (\text{PK}, \text{MK}) \leftarrow \text{Setup}(\text{U}) \).

2. **Enc(\text{PK}, M, Y)**
   - The encryption algorithm takes as input the public parameters \( \text{PK} \), a message \( M \) and an access structure \( Y \) over the universe of the attributes. This algorithm will encrypt \( M \) and produce a ciphertext \( CT \) such that users from the same group will be able to decrypt \( CT \) if the union of their attributes satisfies the access structure \( Y \). We assume the ciphertext implicitly contains \( Y \).
   
   \( CT \leftarrow \text{Enc}(\text{PK}, M, Y) \).

3. **KeyGen(\text{MK}, S, g, \text{PK})**
   - The key generation algorithm takes as input the master key \( \text{MK} \), a group identifier \( g \), an attribute set \( S \) and the public parameters \( \text{PK} \). It outputs a private key \( SK_S \).
   
   \( SK_S \leftarrow \text{KeyGen}(\text{MK}, S, g, \text{PK}) \).

4. **Decrypt(\text{PK}, \text{CT}, U^h, SK_{S_i})**
   - The decryption algorithm takes as input the public parameters \( \text{PK} \), a ciphertext \( CT \) that was encrypted under an access structure \( Y \), and a set of users (each with an attribute set \( S_i^h \), \( i = 1, 2, \ldots, N \)) from the same group \( g \). Let \( SK_{S_i}^h \) be the private key of each user of attribute set \( S_i^h \). The algorithm can
decrypt the CT and output a message M, if \( U^g \) satisfies the access structure \( Y \), where \( U^g = S^g_1 \cup S^g_2 \cup \ldots \cup S^g_{k_g} \).

\[
M \leftarrow \text{Decrypt}(PK, CT, SK_{U^g}, U^g).
\]

Here, \( SK_{U^g} = \{ SK_{S^g_1}, SK_{S^g_2}, \ldots, SK_{S^g_{k_g}} \} \) is the set of private keys of cooperating users.

### 4.1 Security model for GO-CP-ABE

- **Setup**: The challenger runs the Setup algorithm of the GO-CP-ABE, and gives the public parameters \( PK \) to the adversary \( A \).

- **Phase 1**: The adversary \( A \) makes repeated private key queries. Let each query be \((S_i, g_i)\), where \( S_i \) denotes the attribute set and \( g_i \) denotes the group identifier. The attribute set \( U^g = S^g_1 \cup S^g_2 \cup \ldots \cup S^g_{k_g} \) denotes the union of the attribute sets with the same group identifier \( g_i \) that appeared in \( A \)'s queries.

- **Challenge**: The adversary \( A \) submits two equal length messages \( M_0 \) and \( M_1 \). In addition, the adversary \( A \) selects a challenge access structure \( Y^* \) such that none of the union attribute sets \( U^g \) from Phase 1 satisfies the access structure \( Y^* \). The challenger flips a random coin \( \beta \), and encrypts \( M_{\beta} \) under \( Y^* \). The ciphertext \( CT^\beta \) is given to the adversary \( A \).

- **Phase 2**: Phase 1 is repeated with the restriction that none of the union attribute sets \( U^g \) satisfies the access structure \( Y^* \).

- **Guess**: The adversary \( A \) outputs a guess \( \beta' \) of \( \beta \).

The advantage of the adversary \( A \) in this game is defined as \( |\Pr[\beta' = \beta] - \frac{1}{2}| \).

**Definition 4.1**: A GO-CP-ABE scheme is secure in the above security model if all polynomial time adversaries have at most a negligible advantage in above game.

### 5 Our construction

Recall that we want to design an ABE scheme in which a ciphertext encrypted under an access structure \( Y \) can be decrypted only by users from the same group \( g \) if the union of their attribute set \( U^g \) satisfies the access structure \( Y \). Users from different groups cannot make it. So our scheme provides the security against collusion attack.

As described previously, in order to embody the concept of group collaboration, the basic method of our construction is to add a group identifier to reflect this feature. In order to achieve this goal, we modify the existing scheme (Waters, 2011) instead of building a new ABE scheme from scratch. When producing private keys, the authority associates a random value \( \ell_g \) for each user from the same group \( g \). Below we give a description of our construction. Let \( G_1 \) be a bilinear group of prime order \( p \), and let \( g \) be a generator of \( G_1 \). In addition, let \( e: G_1 \times G_1 \rightarrow G_1 \) denote the bilinear map. A security parameter, \( k \), will determine the size of the group. In our construction, the encryption algorithm will take as input a LSSS access matrix \( \mathcal{M} \) and distribute a random exponent \( s \in \mathbb{Z}_p \). Our construction as follows:

- **Setup**: This is a randomised algorithm that takes as input the number of attributes in the system. It then chooses a bilinear group \( G_1 \) of prime order \( p \) with generator \( g \) and \( U \) random group elements \( h_1, \ldots, h_U \in G_1 \) that are associated with the \( U \) attributes in the system. In addition, it will choose two random exponents \( \alpha, \beta \in \mathbb{Z}_p \).

The published public parameters are:

\[
PK = g, e(g, g)^\alpha, g^\beta, h_1, \ldots, h_U.
\]

The master secret key is:

\[
MK = g^\alpha.
\]

- **Enc**\((PK, M, (\mathcal{M}, \rho))\): The encryption algorithm first takes as input the public parameters \( PK \), a message \( M \) and the access structure \( (\mathcal{M}, \rho) \). \( \mathcal{M} \) is a \( l \times n \) matrix converted from the access structure \( Y \) and \( \rho \) is a function that associates each row of the \( \mathcal{M} \) with an attribute. Next, the algorithm randomly selects an exponent \( s \), then chooses a random vector \( \mathbf{v} = (x_1, y_2, \ldots, y_n) \in \mathbb{Z}_p^n \). These values will be used to share the encryption exponent \( s \). For \( i = 1 \), it calculates \( \lambda_i = \mathbf{v} \cdot \mathbf{M}_i \), where \( \mathbf{M}_i \) is the vector corresponding to the \( i \)-th row of \( \mathcal{M} \). Finally, the algorithm chooses randomly \( n_1, \ldots, n_l \in \mathbb{Z}_p \).

The ciphertext \( CT \) is published as follows:

\[
CT = (C = Me(g, g)^\lambda, C_0 = g^\rho, (C_1, D_1), \ldots, (C_l, D_l)).
\]

Here, for \( i = 1, \ldots, l \), \( C_i = g^{\rho h_i n_i}, D_i = g^\alpha \).

- **KeyGen**\((MK, S^g, \ell_g, PK)\): To generate a private key for the attribute set \( S^g \) from the group \( g \), randomly chooses a \( \ell_g \in \mathbb{Z}_p \) for a group \( g \), then takes an input the master key \( MK \) and public parameters \( PK \). It creates the private key as follows:

\[
K = g^{\alpha \ell_g x}, L = g^{\ell_g}, \quad \forall x \in S^g \quad K_x = h_i^{\ell_g}.
\]

- **Dec**\((PK, CT, U^g, SK_{U^g})\): Given a ciphertext, \( CT \), encrypted under an access structure \( (\mathcal{M}, \rho) \) and the union of user attributes \( U^g \), where \( U^g = S^g_1 \cup S^g_2 \cup \ldots \cup S^g_{k_g} \). Let \( SK_{S^g_i} \) denotes the private key of attribute set \( S^g_i \). So \( SK_{U^g} \) is the private key of attribute set \( U^g \). Suppose that \( U^g \) satisfies the access structure and let \( I \subset \{1, 2, \ldots, l\} \) be \( I = \{i: \rho(i) \in U^g\} \). Then, let \( \{\omega_i \in \mathbb{Z}_p\}_{i \in I} \) be a set of constants such
that if \( \lambda, \) are valid shares of any secret \( s \) according to \( M, \) then \( \sum_{i=1}^{k_n} a_i \lambda_i = s. \)

The ciphertext can be decrypted by computing the following:

\[
e(C_0, K)e\left(\prod_{i=1}^{n} C_{i}^{-a_i}, L\right)\prod_{i=1}^{n} e(D_{i}, K_{\rho_0})^{-a_i}
\]
\[
= e\left(g^a, g^{a'}, g^{e_k}\right)\prod_{i=1}^{n} e\left(g^{a_i}, h^{\rho_0}, g^{h_i}\right)\prod_{i=1}^{n} e\left(g^{e_k}, h^{\rho_0}\right)^{-a_i}
\]
\[
= e(g, g)^{e_k}
\]

The decryption algorithm can then divide out this value from \( C \) and obtain the message \( M. \)

6 Proof of security

In this section, we give a security proof of our scheme in the selective security model. The security of our scheme can be reduced to the hardness of decisional \( q \)-parallel BDHE problem.

Theorem 1: Suppose the decisional \( q \)-parallel BDHE assumption holds. Then no polynomial-time adversary can selectively break our scheme with a challenge matrix of size \( \ell \times n', \) where \( \ell, n' \leq q. \)

Proof: Suppose there is an adversary \( \mathcal{A} \) with non-negligible advantage \( \epsilon = \text{Adv}_\mathcal{A} \) in the selective security model against our construction. Additionally, suppose it selects a challenge matrix \( M^* \) where both dimensions are at most \( q. \) Below, we build an algorithm to solve the decisional \( q \)-parallel BDHE problem by running \( \mathcal{A} \) as the subroutine algorithm.

Init. The simulator \( \mathcal{B} \) takes in a \( q \)-parallel BDHE challenge \( \tilde{y}, T. \) The adversary selects the challenge access structure \( (M^*, \rho^*), \) where \( M^* \) has \( n' \) columns.

Setup. The simulator randomly chooses \( a' \in \mathbb{Z}_p \) and implicitly sets \( a = a' + a^{x+1} \) by letting \( e(g, g)^a = e(g^{a'}, g^{e_k}e(g, g)^{a'}). \)

The simulator sets the elements \( h_1, \ldots, h_u \) as follows.

For each \( x \) for \( 1 \leq x \leq U \) begin by choosing a random value \( z_x. \) Let \( X \) denote the set of indices \( i, \) such that \( \rho^*(i) = x. \) The simulator sets \( h_x \) as:

\[ h_x = g^{z_x} \prod_{i \in X} g^{z_x h_i} g^{x^2 z_x h_i} \cdots g^{x^{x^{k_n-1}} z_x h_i}. \]

Note that if \( X = \emptyset \) then we have \( h_x = g^{z_x}. \) All the parameters are distributed randomly due to the random value \( z_x. \)

Phase 1. \( \mathcal{A} \) issues queries for private keys for any attribute sets. Suppose the adversary \( \mathcal{A} \) makes a request for secret key for an attribute set \( S_i \) in the group \( g. \) The simulator chooses a random \( r \in \mathbb{Z}_p. \) Then, it finds a vector \( \tilde{v} = (w_1, \ldots, w_u) \in \mathbb{Z}_p \) such that \( w_1 = -1 \) and for all \( i \) where \( \rho^*(i) = S_i \), we have that \( h_i g_i = 0. \) Such a vector must exist according to the definition of LSSS.

The simulator implicitly defines the same \( t_i \) for users in group \( g \) as

\[ r + w_1 a^x = w_2 a^{x+1} + \ldots + w_u a^{x^{k_n-1}}. \]

It is performed by setting \( L \) as follows:

\[ L = g^r \prod_{i=1,1 \leq i \leq n'} \left(g^{a^{i+x^{i-1}}}\right)^{w_i} = g^{e_k}. \]

According to the definition of \( t_i, \) we have that \( g^{a^{e_k}} \) contains a term of \( g^{a^{x^{k_n-1}}} \), which will cancel out with the unknown term in \( g^e \) when creating \( K. \) The simulator \( \mathcal{B} \) calculates \( K \) as follows:

\[ K = g^{a^e} g^{e_k} \prod_{i=1,1 \leq i \leq n'} \left(g^{a^{i+x^{i-1}}}\right)^{w_i}. \]

Next, we compute \( K_x \) for \( \forall x \in S_i. \) First, for those \( x \in S_i, \) but there is no \( i \) such that \( \rho^*(i) = x, \) we simply let \( K_x = L^e. \)

But for those \( x \in S_i, \) where \( x \) is used in the access structure, the task to create key components \( K_x \) is more difficult. Since we must make sure that there are no terms of the form \( g^{a^{i+x^{i-1}}} \) that we cannot simulate. However, we have that \( h_i g_i = 0. \) For this reason, all of these terms cancel.

Let \( X \) be the set of all \( i \) such that \( \rho^*(i) = x. \) The simulator computes \( K_x \) as follows:

\[ K_x = L^e \prod_{i=1, i \in X} \left(g^{a^{i+x^{i-1}}}\right)^{w_i}. \]

According to the above process, the simulator \( \mathcal{B} \) can calculate private key for the attribute set \( S_i. \)

Challenge. The adversary gives two messages \( M_{0}, M_{1} \) to the simulator \( \mathcal{B}. \) The simulator \( \mathcal{B} \) flips a coin \( \beta. \) It calculates \( C = M_{\beta} Te\left(g^{a}, g^{e_k}\right) \) and \( C' = g^e. \)

The difficult part is to simulate \( C_i \) values since it contains term that we must cancel out. However, the simulator can choose the secret splitting, such that these cancel out. Intuitively, the simulator \( \mathcal{B} \) will choose random \( y_1^*, \ldots, y_i^*, \) and share the secret using the vector \( \tilde{v} = (s, sa^i, y_1^*, \ldots, sa^{x^{k_n-1}} + y_i^*) \in \mathbb{Z}_p. \)

Moreover, the simulator chooses random values \( n_i^*, \ldots, n_{\ell}^* \in \mathbb{Z}_p. \)

For \( i = 1, \ldots, n^*, \) we define \( R_i \) as the set of all \( k \neq i \) such that \( \rho^*(i) = \rho^*(k). \) That is, the set of all other row indices that have the same attribute as row \( i. \) Then the challenge ciphertext components are computed as follows:

\[ D_i = g^{-\tilde{v}} g^{-bh}, C_i = L_i K_i. \]

Here,
\[ L_i = (g^{h_i})^{-\frac{1}{\gamma_i}} \left( \prod_{k \in K} \prod_{j=1}^{\gamma_i} (g^{(h_i(b_i)/h_j)})^{\mu_{ij}} \right) \]

\[ K_i = h_{\rho(i)}^i \left( \prod_{j=2}^{\gamma_i} (g^e)^{\mu_{ij}} \right) \]

Phase 2. The simulator repeats Phase 1 under the same condition as Phase 1.

**Guess.** The adversary \( A \) will submit a guess \( \beta' \) of \( \beta \). If \( \beta = \beta' \) the simulator \( B \) will output \( \mu = 0 \) to indicate that \( T = e(g, g)^{e^{\gamma_1}} \). Otherwise, it will output \( \mu = 1 \) to denote that \( T \) is a random group element in \( \mathbb{G}_2 \).

If \( \mu = 0 \) then the adversary sees an encryption of \( M_\beta \).

The advantage in this situation is \( \text{Adv}_A \) by definition. Therefore, we have \( PR[\{0\} = 1] \) as follows:

\[
PR[\{0\} = 1] = \frac{1}{2} + \text{Adv}_A.
\]

If \( \mu = 1 \), the adversary gains no information about \( \beta \). Consequently, we have \( PR[\{0\} = 1] = \frac{1}{2} \). Therefore, \( B \) can win the decisional \( q \)-parallel BDHE game with non-negligible advantages.

### 7 Complexity analysis

In this section, we aim to compare with (Li et al., 2014) from function and analyse the efficiency of the proposed GO-CP-ABE scheme in terms of time complexity.

- **Function comparison.** Compared with scheme (Li et al., 2014) with threshold structure, our scheme supports monotone access tree structure. Specifically, our scheme can provide a fine-grained access control for users. From the view of classification of ABE, our scheme belongs to CP-ABE, and the scheme (Li et al., 2014) belongs to KP-ABE. Both of the schemes support group cooperation. Namely, each user belongs to a certain group; users from the same group can pool their attributes to decrypt ciphertext.

- **Efficiency.** In the scheme of GO-CP-ABE, the number of public key is \( U + 3 \) and the size of the ciphertext is \( 2(1 + l) \). The number of bilinear pairing operations in decryption algorithm is \( k + 2 \), where \( k \) is the minimum number of attributes needed to satisfy the access structure. Below, we will give the concrete time cost of our scheme in encryption and decryption.

#### 7.1 Simulation results

We simulate the performance of GO-CP-ABE by evaluating the time cost in encryption and decryption based on the pairing-based library (version 0.5.12). The details of the platform we use are shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Simulation platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Ubuntu 10.10</td>
</tr>
<tr>
<td>CPU</td>
<td>Pentium(R) Dual-Core</td>
</tr>
<tr>
<td>Memory</td>
<td>2.97 GB RAM</td>
</tr>
<tr>
<td>Hard disk</td>
<td>500 G/5,400 rpm</td>
</tr>
<tr>
<td>Programming language</td>
<td>C</td>
</tr>
</tbody>
</table>

In the experiment, we design three different monotone access tree structures. In the first scenario (the simple one), the access structure is \( \gamma_1: 'A AND(D OR(B AND C))' \). Alice and Bob are in the same group. Specifically, Alice with attributes \{A, B\} and Bob with attributes \{B, C\}. The union of their attributes satisfy the access structure \( \gamma_1 \). Thus, they can pool their attributes to decrypt ciphertext encrypted under \( \gamma_1 \). The experiment shows that the time costs of our scheme in encryption and decryption are 0.129729 s and 0.108481 s. In the second scenario (the complex one), the access structure is \( \gamma_2: '((A OR B) AND(C AND D)) AND(E OR F))' \). Alice with attributes \{A, C\} and Bob with attributes \{D, E\}. The time costs of our scheme in encryption and decryption are 0.193378 s and 0.131719 s. In the third scenario (the most complex one), the access structure is \( \gamma_3: '(((A AND B) OR(C OR D)) AND(E OR F AND(G AND H)))' \). Alice with attributes \{A, B, C\} and Bob with attributes
\{E, G, H\}. The experiment shows that the time costs of our scheme in encryption and decryption are 0.199432 s and 0.159991 s. The simulation results of our scheme are listed in Table 2. The construction of LSSS matrix for \(Y_1\) and \(Y_2\) are shown in Figure 1 and Figure 2, respectively. Figure 3 is the LSSS matrix we used in the experiment. From Table 2 and Figure 3, we can see that a more complex access structure results in a higher time cost.

**Figure 3** LSSS matrices of \(Y_1\), \(Y_2\) and \(Y_3\)

\[
A = \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
1 & 1 & 0 \\
0 & 0 & -1 \\
1 & 1 & 0 \\
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
0 & -1 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
1 & 0 & 1 \\
E & 0 & 0 \\
0 & -1 & 0 \\
\end{bmatrix}
\]

\[
F = \begin{bmatrix}
1 & 1 & 0 \\
0 & -1 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

**Table 2** Time cost in encryption and decryption

<table>
<thead>
<tr>
<th>Access structure</th>
<th>Encryption Decryption Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Y_1): AND(D OR(B AND C))</td>
<td>0.129727 s 0.108481 s 0.238213 s</td>
</tr>
<tr>
<td>(Y_2): ((A OR B)AND(C AND D))AND(E OR F)</td>
<td>0.193378 s 0.131719 s 0.325098 s</td>
</tr>
<tr>
<td>(Y_3): ((A AND B)OR(C OR D))AND(E OR FG)</td>
<td>0.199432 s 0.159991 s 0.369620 s</td>
</tr>
</tbody>
</table>

**8 Conclusions**

In this paper, we proposed the concept of GO-CP-ABE with more expressive formulas. This can be viewed as an improvement over the threshold GO-ABE scheme in Li et al. (2014). Specifically, in our new notion, message is encrypted under an expressive access structure over attributes and users are described by their attributes. In addition, users are divided into different groups and each user belongs to only one group. Users within the same group can merge their attributes to decrypt successfully, if the union of their attributes satisfies the access structure embedded in the ciphertext. But users from different groups cannot decrypt successfully. We also gave a construction by extending an existing CP-ABE scheme in Waters (2011), together with a rigorous security proof.

**Acknowledgements**

This work is supported by National Natural Science Foundation of China (61472083, 61402110, 61363006), Fok Ying Tung Education Foundation (141065), Program for New Century Excellent Talents in Fujian University (JA14067), Distinguished Young Scholars Fund of Fujian (2016J06013), and Fujian Normal University Innovative Research Team (No. IRTL1207).

**References**


