SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **8**:501–509
Published online 21 March 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.997

Research Article

Large universe decentralized key-policy attribute-based encryption

Qi Li[1]*, Jianfeng Ma[1], Rui Li[2], Jinbo Xiong[1] and Ximeng Liu[3]

[1] School of Computer Science and Technology, Xidian University, Xi'an, China
[2] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
[3] School of Telecommunications Engineering, Xidian University, Xi'an, China

ABSTRACT

In multi-authority attribute-based encryption (ABE) systems, each authority manages a different attribute universe and issues the private keys to users. However, the previous multi-authority ABE schemes are subject to such restrictions during initializing the systems: either the attribute universe is polynomially sized and the attributes have to be enumerated or the attribute universe can be exponentially large, but the size of the set of attributes, which will be used in encryption, is not more than a predefined fixed value. These restrictions prevent multi-authority ABE schemes from being deployed in dynamic practice applications. In this paper, we present a large universe decentralized key-policy ABE scheme without such additional limitation. In our scheme, there is no requirement of any central authority. Each attribute authority executes independently from the others and can join or depart the system allodiality. Our system supports any monotone access policy. The proposed scheme is constructed on prime order groups and proved selectively secure in the standard model. To the best of our knowledge, our scheme is the first large universe decentralized key-policy ABE system in the standard model. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
KP-ABE; large universe; multi-authority; decentralized

***Correspondence**
Qi Li, School of Computer Science and Technology, Xidian University, Xi'an South Taibai Road, Shannxi, 710071, China.
E-mail: qilijs@gmail.com

1. INTRODUCTION

In open communication scenarios, one must encrypt the sensitive data before the data are transmitted or stored. Nevertheless, traditional cryptosystems cannot support complex access structures and are useless in such applications, where the recipient is denoted by a set of descriptive attributes rather than a public key or identity. Sahai and Waters [1] gave a solution to this issue by presenting attribute-based encryption (ABE). In the proposed ABE system, the ciphertext is labeled with a set of attributes, a central authority (CA) issues the private keys to each user corresponding to his or her attributes. A user can successfully decrypt the ciphertext if and only if there is an overlap between his or her attributes and the set of attributes in the ciphertext.

Subsequently, Goyal *et al*. proposed the first construction of key-policy ABE (KP-ABE) [2] and further formulated another kind of ABE: ciphertext-policy ABE (CP-ABE). In KP-ABE schemes, the ciphertext is annotated with a set of attributes while the user's private keys are labeled with an access structure. On the contrary, in CP-ABE systems, the ciphertext is associated with an access policy while a set of attributes is attached to the user's private keys. The first CP-ABE system was presented in [3], and more ABE schemes [4–9] are designed since then. However, in these ABE systems mentioned before, only a single CA is supported. These single-authority ABE systems are useless in distributed application environments, where the attributes may be issued and administered by different authorities. To address this problem, Chase [10] proposed the first multi-authority ABE system. From then on, more multi-authority ABE schemes have been constructed [11–14].

A limitation of the prior ABE systems is that, once the system parameters have been selected at the setup phase,

† Please ensure that you use the most up to date class file, available from the SEC Home Page at
http://www3.interscience.wiley.com/journal/114299116/home

these systems cannot offer complete flexibility in choosing the attributes or access policies. Lewko *et al*. [15] first addressed this issue and introduced a classification of ABE schemes: small universe and large universe. In "small universe" ABE schemes (e.g., [3,6,7,10]), the size of attribute universe is polynomial to the system security parameter, and the attributes in the universe must be fixed at the setup stage. The size of system public parameters increases linearly with the amount of attributes in the chosen universe. In "large universe" ABE schemes, the attribute universe can be exponentially large. However, in the "large universe" constructions, either the maximum number of the attributes that will be employed in encryption is limited to a parameter n, which must be fixed at the setup phase, such as the ABE systems in [2,16], or the ABE scheme was proved secure in the random oracle model, such as [3].

Such limitation prevents ABE systems from being deployed in practical applications. For instance, if the system parameters are selected to be too small, the system cannot provide sufficient domain of potential attributes and will have to be re-initialized while the user possesses the attributes that overstep the restriction. If the system parameters are selected to be too large, unnecessary computation overhead will be brought in all operations. To remove such restriction, Lewko *et al*. [15] presented the first large universe ABE system in the standard model. They constructed the KP-ABE system on composite order groups. In their scheme, the number of system public parameters is constant size, but the supported attribute universe is exponential size. Afterwards, Rouselakis *et al*. [17] proposed two large universe ABE schemes (one CP-ABE and one KP-ABE) on prime order groups and proved the security under q-type assumptions in the standard model. Nevertheless, these ABE schemes [15,17] are restricted to only a single authority. To the best of our knowledge, it remains an open problem that how to construct a large universe KP-ABE system in multi-authority setting.

In this paper, we present a large universe decentralized KP-ABE system in the standard model by extending the single-authority KP-ABE system [17] into the multi-authority settings. In our scheme, there is no CA, and any participant can be an attribute authority (AA) by publishing its public parameters and an attribute universe. Because the number of AAs is less small than the amount of attributes, the size of system public parameters goes linearly with the number of AAs is acceptable. Each AA issues private keys to different users and operates entirely independently. To prevent unauthorized users from decrypting the ciphertext by combining their private keys, we link each user's private keys with its global identifier (GID).

The main contributions of this work are summarized as follows:

(1) We present a novel large universe decentralized KP-ABE system on prime order groups. The proposed scheme supports a large universe of attributes and does not impose any bounds on the set of attributes, which will be used in encryption.
(2) We prove the selective security of the proposed scheme under the $q - type$ assumption in the standard model. In addition, our system is secure against at most $F - 1$ AAs corruption, where F denotes the number of AAs in the scheme.
(3) Performance comparisons show that the efficiency of our scheme is comparable to the underlying single-authority KP-ABE scheme.

1.1. Related work

Sahai and Waters [1] first introduced the concept of ABE and left an open problem that whether it is possible to construct an ABE system where the attributes are issued by different authorities. Chase [10] gave an affirmative answer by presenting the first multi-authority KP-ABE system, where a CA and multiple AAs existed. The most challenging problem in multi-authority ABE is to resist the collusion attacks from multiple unauthorized users. Chase [10] resolved this problem by labeling each user with an unique GID. A user's private keys from different AAs will be linked together by his or her GID. Because the CA chose the secret key for each AA, it can decrypt all ciphertexts. Lin *et al*. [18] presented a multi-authority KP-ABE system without any CA employing a threshold technique, where multiple authorities must cooperate during initializing the system. Their system is not secure if k or more users collude, where k is a bound value fixed in the setup phase. Chase *et al*. [11] also removed the need of a CA by applying a distributed pseudorandom function technique. Moreover, they presented an anonymous key issuing protocol for protecting the privacy of users. In this system, the AAs also need to cooperate in the setup phase.

The first multi-authority CP-ABE scheme was presented by Müller *et al*. [13,14]. In this system, there must be a CA who generates the global public parameters, and each AA can execute independently from the others. Lewko and Waters [12] presented a decentralizing CP-ABE system. Being different from all multi-authority ABE schemes aforementioned, which are only selectively secure, this system achieves full security and is proven security in the random oracle model. Moreover, while the systems in [10,11,18] support tree access structures and the systems in [13,14] support the policy written in disjunctive normal form (DNF), the new scheme can support any linear secret sharing scheme (LSSS) access matrix. In this system, there is no requirement of a CA, and multiple AAs operate independently in both the setup and key generation phases. Each participant can be an AA by issuing the public parameters, and each AA can join or depart the system freely without rebuilding the system. Liu *et al.* [16] presented the first fully secure multi-authority CP-ABE system in the standard model. In this system, there are multiple CAs and AAs. The CAs issue the private keys to a user that reflect his or her identity. The AAs manage different attribute universes and issue the private keys to a user that reflect his or her attributes. Neither the CA nor the AA can independently decrypt the ciphertext. Moreover,

DOI: 10.1002/sec

because multiple CAs are employed to prevent some colluding CAs from decrypting the ciphertext, the number of CAs can be small. Li *et al.* [19] presented a multi-authority CP-ABE system with accountability, where the misbehaving user can be traced when he or she leaked his or her private keys to others. However, multiple AAs must collaborate during the system initialization. Table I describes some characteristics of current ABE schemes and ours.

1.2. Organization

In Section 2, we provide some notation of bilinear maps, access structures, LSSSs, and the assumption. Additionally, we introduce the definition of the decentralized KP-ABE and the security game. Section 3 gives the detailed construction of our large universe decentralized KP-ABE system. Section 4 presents the results of performance comparison. Section 5 gives the security proof of the proposed scheme. Finally, we conclude in Section 6.

2. BACKGROUNDS

2.1. Bilinear maps

We use the definition of the bilinear maps from [1,20].

Select two multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_1$ of large prime order p. g is a generator of $\mathbb{G}$. The map e is a bilinear map if e has such properties:

(1) Bilinearity: $\forall h, \theta \in \mathbb{G}$ and $x, y \in \mathbb{Z}_p$, we have $e(h^x, \theta^y) = e(h, \theta)^{xy}$.
(2) Non-degeneracy: $e(g, g) \neq 1$.

The group $\mathbb{G}$ is said to be an admissible bilinear group if the group operations in $\mathbb{G}$ and the bilinear map e can be efficiently computed. Furthermore, e is a symmetric map because $e(g^x, g^y) = e(g^y, g^x) = e(g, g)^{xy}$.

2.2. Access structure

Definition 1. *Access Structure* [21]*: Let* $\mathbb{P} = \{P_1, P_2, \ldots, P_T\}$ *denote a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_T\}}$ *is monotonic if* $\forall A_1, A_2$*: if* $A_1 \in \mathbb{A}$ *and* $A_1 \subseteq A_2$ *then we have* $A_2 \in \mathbb{A}$*. An access structure (respectively, monotonic access structure) is a collection (respectively, monotonic collection)* $\mathbb{A}$ *of non-empty subsets of* $\mathbb{P}$*. That is,* $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_T\}} \backslash \{\emptyset\}$*. We say that the sets in* $\mathbb{A}$ *are the authorized sets, and the sets outside* $\mathbb{A}$ *are the unauthorized sets.*

Among ABE systems, the role of the parties is replaced by the descriptive attributes. In this way, the authorized set of attributes will be contained in the access structure $\mathbb{A}$. We focus on the monotonic access structure in this paper. To realize common access structures, one can simply consider the negation of an attribute as a separate attribute. Moreover, we assume that an attribute is used at most once in the LSSS matrix. Such restriction can be removed by employing the technique in [7].

2.3. Linear secret sharing schemes

Here, we adopt the definition of LSSS from [5,21]:

Definition 2. *Linear secret sharing schemes: Let* $\mathbb{P}$ *be a set of parties, p be a prime. A secret sharing scheme* Π *over* $\mathbb{P}$ *is linear (over* $\mathbb{Z}_p$*) if it has the following properties:*

(1) The shares of a secret for each party form a vector over $\mathbb{Z}_p$*.*
(2) There is a matrix $A \in \mathbb{Z}_p^{\ell \times n}$*, which is called the share-generating matrix for* Π*. For all* $i = 1, \ldots, \ell$*, there exists a function* ρ *that labels the i-th row of A with a party. (i.e.,* $\rho \in \mathcal{F}([\ell] \to \mathbb{P})$*). During generating the shares, we consider the column vector* $\overrightarrow{v} = (s, r_2, \ldots, r_n)^\top$*, where* $s \in \mathbb{Z}_p$ *denotes the secret to be shared, and* $r_2, \ldots, r_n$ *are randomly picked from* $\mathbb{Z}_p$*, then* $A\overrightarrow{v}$ *is the vector of* ℓ *shares of s according to* Π*. The shares* $(A\overrightarrow{v})_i$ *belongs to the party* $\rho(i)$*.*

As shown in [21], each LSSS mentioned before must satisfy the linear reconstruction requirement, defined as follows: Assume that an access structure $\mathbb{A}$ is denoted by (A, ρ). Π is an LSSS for $\mathbb{A}$. Let S denote an authorized set.

Table I. A Comparison between current ABE schemes and ours.

Schemes	CP/KP	Multi-authority	Security	Standard model	Access structure	Supporting large universe	Group order
[10,11]	KP	YES	Selective	YES	Tree	YES (restricted)	Prime
[13,14]	CP	YES	Selective	YES	DNF	YES (restricted)	Prime
[18]	KP	YES	Selective	YES	Tree	YES (restricted)	Prime
[12]	CP	YES	Adaptive	NO	LSSS	NO	Composite
[16]	CP	YES	Adaptive	YES	LSSS	YES (restricted)	Composite
[19]	CP	YES	Selective	YES	AND-gate	YES (restricted)	Prime
[15]	KP	NO	Selective	YES	LSSS	YES	Composite
[17]	KP,CP	NO	Selective	YES	LSSS	YES	Prime
Ours	KP	YES	Selective	YES	LSSS	YES	Prime

Then, let $I = \{i : \rho(i) \in S\}$ denote the index set of rows whose labels are in S. There exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ such that if $\{\lambda_i = (A\overrightarrow{v})_i\}$ are valid shares of a secret s according to Π, then we have $\sum_{i\in I}\omega_i\lambda_i = s$. Moreover, such constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ can be found in time polynomial in the size of the matrix A. Nevertheless, if the set S is unauthorized, no such constants exist.

2.4. q-type assumption

For our large universe decentralized KP-ABE system, we will follow the q-type assumption that was proposed in [17], where the detailed proof can also be found. The assumption is defined by the following game run by an adversary and a challenger:

The challenger selects two groups $\mathbb{G}, \mathbb{G}_1$ of prime order p, picks a generator g of $\mathbb{G}$. It chooses $q + 3$ exponents $x, y, z, b_1, b_2, \ldots, b_q$ randomly from $\mathbb{Z}_p$. A bilinear map e is $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$. The adversary is given the group $(p, \mathbb{G}, \mathbb{G}_1, e)$ and the whole following elements:

$X =$

$g, g^x, g^y, g^z, g^{(xz)^2}$

$g^{b_i}, g^{xzb_i}, g^{xz/b_i}, g^{x^2zb_i}, g^{y/b_i^2}, g^{y^2/b_i^2}, \forall i \in [q]$

$g^{xzb_i/b_j}, g^{yb_i/b_j^2}, g^{xyzb_i/b_j}, g^{(xz)^2b_i/b_j}, \forall i, j \in [q], i \neq j$

Then, the challenger picks a random coin $o \in \{0, 1\}$. If $o = 0$, it sends the term $T = e(g, g)^{xyz}$ to the adversary. Otherwise, it sends $T = R$, where R is a random element from $\mathbb{G}_1$. Finally, the adversary has to output a guess o' on o. We define the advantage of an adversary in solving the decisional q-type problem in $\mathbb{G}$ as $Adv = Pr[o' = o] - 1/2$.

Definition 3. *We say that the q-type assumption holds if no PPT adversary has a non-negligible advantage in resolving the earlier security game.*

2.5. Decentralized KP-ABE

There are multiple AAs and users in a decentralized KP-ABE scheme. We let $\mathbb{F} = \{1, 2, \ldots, F\}$ be the index set of the AAs. That is, $f \in \mathbb{F}$ denotes the index of AA_f. Each user is associated with a unique fixed GID. Each AA administers a different attribute universe from the others.

We let U_f be the attribute universe administered by AA_f. For all $i \neq j \in \mathbb{F}$, we have $U_i \bigcap U_j = \emptyset$. We let $U = \bigcup_{f=1}^{F} U_f$ denote the total attribute universe in the system.

A multi-authority KP-ABE system is composed of the following 5 algorithms:

GlobalSetup$(\lambda) \to (GPK)$: This algorithm takes in a security parameter λ. It outputs the global system parameters GPK.

AASetup $(GPK, f, U_f) \to (AAPK_f, AAMSK_f)$: Each AA_f runs this algorithm to generate its public parameter $AAPK_f$ and the corresponding master secret key $AAMSK_f$.

Encrypt $(M, S, GPK, \bigcup AAPK_f) \to (CT)$: This algorithm takes in the global public parameters GPK, a set of attributes S, a message M, and the set of public parameters $\bigcup AAPK_f$ for relevant AAs. It then outputs a ciphertext CT. We assume the set S is implicitly included in CT.

AAKeyGen $(\mathbb{A}_{GID,f}, GPK, AAMSK_f) \to (UAASK_{\mathbb{A}_{GID,f}})$: Each AA_f runs this algorithm by taking in a user's GID, an access structure $\mathbb{A}_{GID,f}$, GPK, and the master secret key $AAMSK_f$, where $\mathbb{A}_{GID,f}$ is expressed by an LSSS matrix $(A_{GID,f}, \rho)$. It then gives the private key $UAASK_{\mathbb{A}_{GID,f}}$ to the user. We assume the access structure $\mathbb{A}_{GID,f}$ is implicitly included in $UAASK_{\mathbb{A}_{GID,f}}$.

Decrypt $(CT, GPK, GID, \bigcup UAASK_{\mathbb{A}_{GID,f}}) \to (M)$: This algorithm takes in CT, GPK, GID, and $\bigcup UAASK_{\mathbb{A}_{GID,f}}$, where all the user private keys are labeled with the same GID. If the set of attributes S satisfies the access structure $\mathbb{A}_{GID}$, where $\mathbb{A}_{GID} = \bigcup \mathbb{A}_{GID,f}$, the algorithm outputs M, otherwise, it outputs $\perp$.

2.6. Selective security game

Our security model for the multi-authority ABE is similar to the game introduced in [10,22,23], where the adversary must declare the challenge set of attributes before the setup phase. In addition, we suppose that the adversary is allowed to corrupt the AAs only statically. The detailed definition of our selective security game is given as follows:

Initialization: The adversary specifies a index set of corrupt AAs, which is denoted by $\mathbb{F}_c$. We let $\mathbb{F}_{uc} = \mathbb{F} \backslash \mathbb{F}_c$ be the index set of uncorrupt AAs. Additionally, the adversary submits a set of attributes S^*, which he or she wants to challenge in the security game. In order to facilitate understanding, we write $S^* = \bigcup S_f^*$, where each S_f^* is managed by the attribute authority AA_f.

Setup: The simulator runs the **GlobalSetup** algorithm to provide the public parameters GPK. In **AASetup** phase, for each corrupt AA, the simulator produces $AAPK_f$ and $AAMSK_f$ and passes them to the adversary. For each uncorrupt AA, only the public parameter $AAPK_f$ is sent to the adversary.

Phase 1: The adversary $\mathcal{A}$ can query the secret keys for each GID as follows:

To answer the key queries on the access structures that belongs to the corrupt AAs, the secret keys can be generated by the adversary itself. In contrast, the simulator will answer the key queries on the access structures belonging to the uncorrupt AAs. These types of queries can be made adaptively other than such restriction that, for all $f \in \mathbb{F}_{uc}$, at least a set S_f^* cannot satisfy the chosen access structure issued by AA_f.

Challenge: The adversary $\mathcal{A}$ submits two massages M_0 and M_1 with equal length. $\mathcal{B}$ flips a random coins $\mu \in \{0, 1\}$ and encrypts M_μ under S^*. It then gives the ciphertext CT^* to $\mathcal{A}$.

Phase 2: The adversary repeats the queries as in Phase 1.

Guess: $\mathcal{A}$ outputs a guess μ' on μ. The advantage of $\mathcal{A}$ is defined as $Pr[\mu' = \mu] - 1/2$.

Definition 4. *A multi-authority KP-ABE system is selectively secure if all PPT adversaries have advantage at most negligible in the earlier game.*

For simplifying the description of the scheme, we adopt the assumption proposed in the systems [10,16]. That is, a user's GID is assumed to request the corresponding private keys from each AA only once in our scheme. To remove the assumption, one can add a time stamp on GID [16] or give the user a new GID [10].

3. LARGE UNIVERSE DECENTRALIZED KP-ABE SCHEME

We now show how to construct a large universe decentralized KP-ABE scheme, which is secure against at most $F-1$ AAs corruption, where F denotes the number of AAs in the system. The idea is encouraged by the single-authority KP-ABE scheme [17] and the multi-authority ABE systems [10–12]. The detailed construction is given in the following way:

GlobalSetup$(\lambda)\rightarrow$ (GPK): This algorithm takes in the security parameter λ and publishes the terms $(p, \mathbb{G}, \mathbb{G}_1, e)$, where $\mathbb{G}$ and $\mathbb{G}_1$ are the bilinear groups of prime order p, and e is a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$. Let g be a generator of $\mathbb{G}$. The algorithm then selects θ, h, ω, ν randomly from $\mathbb{G}$. The global public parameters is $GPK = (p, \mathbb{G}, \mathbb{G}_1, e; g, \theta, h, \omega, \nu)$.

AASetup $(GPK, f, U_f) \rightarrow (AAPK_f, AAMSK_f)$: Each AA_f first picks two random exponents $\alpha_f, \beta_f \in \mathbb{Z}_p$. It then computes $AAPK_{f,1} = e(g,g)^{\alpha_f}$ and $AAPK_{f,2} = g^{\beta_f}$. Finally, it sets the public parameter $AAPK_f = \left(AAPK_{f,1}, AAPK_{f,2}\right)$ and keeps α_f and β_f as its master secret keys.

Encrypt $\left(M, S, GPK, \bigcup AAPK_f\right) \rightarrow (CT)$: We denote the set of attributes S as $S = \bigcup S_f = \{ATT_1, ATT_2, \ldots, ATT_m\}$, where each S_f belongs to the corresponding authority AA_f and $m = |S|$ is the number of attributes in the set S. The encryption algorithm first selects $m+1$ exponents $s, r_1, r_2, \ldots, r_m$ randomly from $\mathbb{Z}_p$. It then computes $C = M \cdot \prod_{f\in\mathbb{F}_E} e(g,g)^{\alpha_f s}$, $C_0 = g^s$, $C_1 = \prod_{f\in\mathbb{F}_E} g^{\beta_f s}$, where $\mathbb{F}_E$ denotes the index set of the corresponding AAs. And for each $k \in [m]$, it calculates $C_{2,k} = g^{r_k}$ and $C_{3,k} = (\theta^{ATT_k} h)^{r_k} \omega^{-s}$. The ciphertext CT is published as $CT = \left(C, C_0, C_1, \{C_{2,k}, C_{3,k}\}_{k\in[m]}\right)$

AAKeyGen $(\mathbb{A}_{GID,f}, GPK, AAMSK_f) \rightarrow \left(UAASK_{\mathbb{A}_{GID,f}}\right)$: For each $f \in \mathbb{F}$, the algorithm takes in a user's $GID \in \mathbb{Z}_p$ and an $\ell_f \times n_f$ access structure $\mathbb{A}_{GID,f} = (A_{GID,f}, \rho)$. It works in the following way: It first selects a vector $\overrightarrow{\upsilon_f} = \left(\alpha_f, \upsilon_2, \ldots, \upsilon_{n_f}\right)^\top$ where $\upsilon_2, \ldots, \upsilon_{n_f}$ are randomly chosen from $\mathbb{Z}_p$. The master secret key α_f will be shared by computing $\overrightarrow{\lambda_f} = \left(\lambda_1, \lambda_2, \ldots, \lambda_{\ell_f}\right)^\top = A_{GID,f} \cdot \overrightarrow{\upsilon_f}$. Meanwhile, the algorithm chooses another vector $\overrightarrow{\psi_f} = \left(GID, \psi_2, \ldots, \psi_{n_f}\right)^\top$ where $\psi_2, \ldots, \psi_{n_f}$ are randomly selected from $\mathbb{Z}_p$ and computes $\overrightarrow{\phi_f} = \left(\phi_1, \phi_2, \ldots, \phi_{\ell_f}\right)\top = A_{GID,f} \cdot \overrightarrow{\psi_f}$. It then chooses ℓ_f random exponents $t_1, t_2, \ldots, t_{\ell_f}$ from $\mathbb{Z}_p$. For each $k \in \left[\ell_f\right]$, it calculates $K_{1,k,f} = g^{\lambda_k} \omega^{t_k} \nu^{\phi_k \beta_f}$, $K_{2,k,f} = \left(\theta^{\rho(k)} h\right)^{-t_k}$ and $K_{3,k,f} = g^{t_k}$.

The user's private keys are published as $UAASK_{\mathbb{A}_{GID,f}} = \{K_{1,k}, K_{2,k}, K_{3,k}\}_{k\in[\ell_f]}$

Decrypt $\left(CT, GPK, GID, \bigcup UAASK_{\mathbb{A}_{GID,f}}\right) \rightarrow (M)$: To decrypt the ciphertext, the decryption algorithm first checks whether the set S in the ciphertext satisfies the access structures $\bigcup UAASK_{\mathbb{A}_{GID,f}}$ in the private keys. If so, for each set S_f, there must be some constants $\{c_{i,f} \in \mathbb{Z}_p\}_{i\in I_f}$, which satisfy that $\sum_{i\in I_f} c_{i,f} A_{i,f} = (1, 0, \ldots, 0)$, where $I_f = \{i : \rho(i) \in S_f\}$ and $A_{i,f}$ is the i-th row of $A_{GID,f}$. These constants can be found in polynomial time if the set S_f satisfies the access structure. Then, it computes

$$B = \frac{\prod_{f\in\mathbb{F}_E} \prod_{i\in I_f} \left(e(C_0, K_{1,i,f})\, e(C_{2,i}, K_{2,i,f})\, e(C_{3,i}, K_{3,i,f})\right)^{c_{i,f}}}{e(\nu^{GID}, C_1)}$$

Finally, the algorithm calculates $M = C/B$

Correctness: If S_f in the ciphertext is an authorized set, we have $\sum_{i\in I_f} c_{i,f}\lambda_{i,f} = \alpha_f$ and $\sum_{i\in I_f} c_{i,f}\phi_{i,f} = GID$. Thereby,

$$\begin{aligned}
&\prod_{f\in\mathbb{F}_E} \prod_{i\in I_f} \left(e(C_0, K_{1,i,f})\, e(C_{2,i}, K_{2,i,f})\, e(C_{3,i}, K_{3,i,f})\right)^{c_{i,f}} \\
&\quad = \prod_{f\in\mathbb{F}_E} \prod_{i_f\in I_f} e(g,g)^{s\lambda_{i,f} c_{i,f}}\, e(g,\nu)^{s\beta_f \phi_{i,f} c_{i,f}} \\
&\quad = \prod_{f\in\mathbb{F}_E} e(g,g)^{s\alpha_f}\, e(g,\nu)^{s\beta_f GID}
\end{aligned}$$

$$e\left(\nu^{GID}, C_1\right) = e\left(\nu^{GID}, \prod_{f\in\mathbb{F}_E} g^{\beta_f s}\right) = \prod_{f\in\mathbb{F}_E} e(\nu, g)^{s\beta_f GID}$$

Then, we have

$$B = \prod_{f\in\mathbb{F}_E} e(g,g)^{\alpha_f s}$$

4. PERFORMANCE ANALYSIS

Table II compares the single-authority KP-ABE system [17], the large universe construction of multi-authority KP-ABE system [10], the decentralizing CP-ABE scheme [12], and our KP-ABE scheme. The sizes of the system public parameters PK, ciphertext, and private keys are calculated in terms of the amount of group elements. In this table, S is the set of attributes in the ciphertext. ℓ is the

Table II. Performances comparison.

	RW' scheme [17]	Chase's scheme [10]	LW's scheme [12]	Our scheme
Ciphertext size	$2\|S\|+2$	$\|S\|+2$	$3\|S\|+1$	$2\|S\|+3$
Private key size	3ℓ	$2\ell+1$	ℓ	3ℓ
decryption pairs	$3I$	$2I+1$	$2I$	$3I+1$
Size of PK	5	$\sum_{f=1}^{F}(n_f+1)+2$	$2\|U\|$	$2F+4$

number of attributes in the user's private keys. I is the number of attributes in S that are utilized in decryption. We let U be the attribute universe in the system and F be the total number of authorities. n_f denotes the maximum number of attributes administered by each AA_f that can be used in encryption.

In [10], the threshold tree access structures are supported, while the LSSS matrix policies are employed in [12,17] and our scheme. All of the schemes are selectively secure except the fully secure decentralizing CP-ABE scheme [12], which is proven security in the random oracle model. In some resource-limited practical application, selective security can be a considerable trade off for efficiency. The size of public parameters of the frameworks [10,12] goes linearly with the amount of attributes in the universe. However, in [17] and our scheme, the size of the system public parameters is related to the number of AAs rather than the size of the attribute universe (especially, there is only an authority in the KP-ABE system [17], the number of public parameters is a constant value). Because that the number of authorities is less small than that of attributes, our scheme is efficient in producing the system public parameters. In addition, each authority can join or depart without rebuilding the system. Hence, our scheme is rather desirable in distributed and dynamic practice applications.

We implement RW's system [17] and ours on a PC with an Intel Core 2 Quad CPU at 2.83 GHz and 4.00 GB RAM. The implementation employs the pairing-based cryptography library with version 0.5.14 [24]. We use a 160-bit Type A elliptic curve group to provide public parameters on 80-bit security level. Such parameter settings are also employed in the system [3]. All the implementation results are the average of 200 trials.

Figure 1 shows the comparisons of computation cost of encryption and decryption. The number of authorities of our scheme is set to be 5. Figure 1(a) describes the comparison of encryption time with different number of attributes. Figure 1(b) shows the comparison of decryption time versus the number of attributes, which are used in decryption. We can see that the encryption and decryption overhead of both RW's system and ours go linearly with the number of attributes in ciphertext and decryption, respectively. Furthermore, the time cost of encryption and decryption in our scheme is just a little more than that in RW's system. Indeed, our scheme performs almost the same result with the underlying single-authority KP-ABE scheme [17].

5. SECURITY ANALYSIS

Theorem 1. *Suppose the q-type assumption holds, then no PPT adversary has non-negligible advantage in selectively breaking our scheme with a challenge set of attributes of size t, where* $t \leq q$.

Assume that there is a PPT adversary $\mathcal{A}$ who can break our large universe multi-authority KP-ABE system with a non-negligible advantage, then we can use $\mathcal{A}$ to construct a simulator $\mathcal{B}$, which can solve the q-type assumption with a non-negligible advantage. The detailed construction of the security game is given as follows:

Initialization: The simulator $\mathcal{B}$ receives a group of elements from the q-type assumption. The adversary $\mathcal{A}$ submits the challenge set of attributes $S^* = \bigcup S_f^* = \{ATT_1^*, ATT_2^*, \ldots, ATT_t^*\}$ and an index set $\mathbb{F}_c$ of the corrupt AAs. We let $\mathbb{F}_{uc} = \mathbb{F} \setminus \mathbb{F}_c$ be the index set of uncorrupt

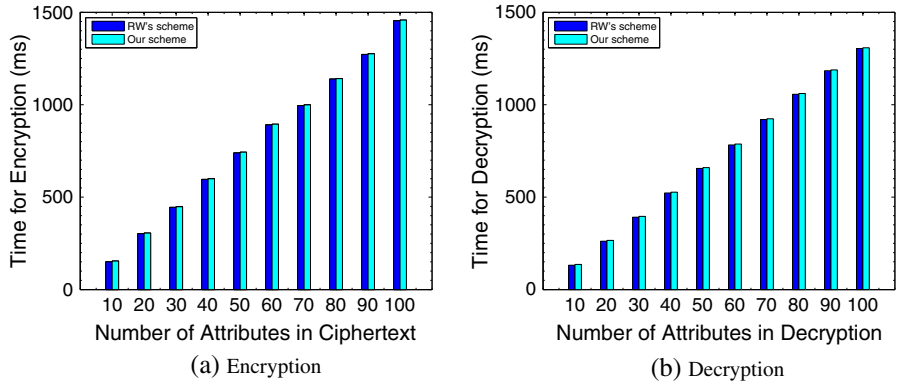

Figure 1. Comparisons of computation cost with different number of attributes. (a) Encryption and (b) decryption.

DOI: 10.1002/sec

AAs. Without loss of generality, we suppose the attribute authority AA_{f*} be the only one uncorrupt authority. Additionally, we assume that S^* is related to all the AAs in the system.

GlobalSetup: The simulator $\mathcal{B}$ produces the global public parameters in the following way:

$$\begin{aligned} g &= g \\ \theta &= g^a \cdot \prod_{i\in[t]} g^{y/b_i^2} \\ h &= g^b \cdot \prod_{i\in[t]} g^{xz/b_i} \cdot \prod_{i\in[t]} \left(g^{y/b_i^2}\right)^{-ATT_i^*} \\ \omega &= g^x \\ \nu &= g^c \end{aligned}$$

where a, b, c are randomly selected from $\mathbb{Z}_p$. We note that ω is a properly random element from $\mathbb{G}$ in the adversary's view. Additionally, the terms θ, ω, and ν are properly distributed because of the random chosen a, b, c.

AASetup: To set the public parameters for the corrupt AAs, the simulator operates as follows: For each $f \in \mathbb{F}_c$, it selects random exponents $\alpha_f, \beta_f \in \mathbb{Z}_p$ and computes $AAPK_{f,1} = e(g,g)^{\alpha_f}$ and $AAPK_{f,2} = g^{\beta_f}$. It then sends $AAPK_{f,1}, AAPK_{f,2}$ and α_f, β_f to the adversary.

To set the public parameters for AA_{f*}, the simulator sets $\alpha_{f*} = xy - \sum_{f\in\mathbb{F}_c} \alpha_f$ and computes $AAPK_{f*,1} = e(g^x, g^y) \prod_{f\in\mathbb{F}_c} e(g,g)^{-\alpha_f} = e(g,g)^{xy-\sum_{f\in\mathbb{F}_c}\alpha_f}$ and $AAPK_{f*,2} = g^{\beta_{f*}}$, where β_{f*} is a random exponent chosen from $\mathbb{Z}_p$.

Phase 1: The adversary makes the secret key query by submitting a user's GID to the simulator with a set of access matrices, where each access matrix is issued and managed by an AA. The query is answered as follows:

For the corrupt AAs, the adversary can make the secret keys on any access matrix.

If the GID is submitted to AA_{f*} along with an access structure $\mathbb{A}_{GID,f*} = (A_{GID,f*}, \rho)$, where $A_{GID,f*}$ is a matrix with ℓ rows and n columns. The simulator $\mathcal{B}$ responds in the following way:

Because S^* is not an authorized set and AA_{f*} is the only one uncorrupt authority, the set S^*_{f*} cannot satisfy the matrix $\mathbb{A}_{GID,f*}$. In this way, there must be a vector $\overrightarrow{\eta} = (1, \eta_2, \ldots, \eta_n)^\top \in \mathbb{Z}_p$ such $\langle A_d, \overrightarrow{\eta}\rangle = 0$ for all $d \in [\ell]$ that $\rho(d) \in S^*_{f*}$. The vector $\overrightarrow{\upsilon_{f*}}$ can be denoted as

$$\overrightarrow{\upsilon_{f*}} = \left(xy - \sum_{f\in\mathbb{F}_c} \alpha_f\right)\overrightarrow{\eta} + \left(0, \upsilon'_2, \ldots, \upsilon'_n\right)^\top$$

where the exponents $\upsilon'_2, \ldots, \upsilon'_n$ are randomly chosen from $\mathbb{Z}_p$. We note that the first component of $\overrightarrow{\upsilon_{f*}}$ is $\alpha_{f*} = xy - \sum_{f\in\mathbb{F}_c}\alpha_f$ and the other components are uniformly random from $\mathbb{Z}_p$. Thereby, for each $d \in [\ell]$, the share of α_{f*} is

$$\begin{aligned} \lambda_d &= \langle A_d, \overrightarrow{\upsilon_{f*}}\rangle \\ &= xy\langle A_d, \overrightarrow{\eta}\rangle - \textstyle\sum_{f\in\mathbb{F}_c}\alpha_f\langle A_d, \overrightarrow{\eta}\rangle \\ &\quad + \langle A_d, \left(0, \upsilon'_2, \ldots, \upsilon'_n\right)^\top\rangle \\ &= xy\langle A_d, \overrightarrow{\eta}\rangle + \lambda'_d \end{aligned}$$

where A_d denotes the d-th row of the matrix $A_{GID,f*}$. In addition, the vectors $\overrightarrow{\psi_{f*}}$ and $\overrightarrow{\phi_{f*}}$ are set as in the AAKeyGen algorithm.

For each $d \in [\ell]$ that $\rho(d) \in S^*_{f*}$, we have $\langle A_d, \overrightarrow{\eta}\rangle = 0$. Therefore, the simulator can pick random exponent $t_d \in \mathbb{Z}_p$ and compute $UAASK_{\mathbb{A}_{GID,f*}}$ as in the real **AAKeyGen** algorithm.

For each $d \in [\ell]$ that $\rho(d) \notin S^*_{f*}$, the simulator selects random exponent $t'_d \in \mathbb{Z}_p$ and sets

$$t_d = -y\langle A_d, \overrightarrow{\eta}\rangle + \sum_{i\in[t]} \frac{xzb_i\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d) - ATT_i^*} + t'_d$$

Hence, the simulator can compute $UAASK_{\mathbb{A}_{GID,f*}}$ for such row d using the terms from the assumption:

$$\begin{aligned} K_{1,d,f*} &= g^{\lambda_d}\omega^{t_d}\nu^{\phi_d\beta_{f*}} \\ &= g^{-xy\langle A_d, \overrightarrow{\eta}\rangle + \sum_{i\in[t]} \frac{x^2zb_i\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d)-ATT_i^*}} \\ &\quad \cdot g^{xy\langle A_d, \overrightarrow{\eta}\rangle + \lambda'_d} \cdot \omega^{t'_d} \cdot \nu^{\phi_d\beta_{f*}} \\ &= g^{\lambda'_d} \cdot \prod_{i\in[t]} \left(g^{x^2zb_i}\right)^{\frac{\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d)-ATT_i^*}} \cdot \omega^{t'_d} \cdot \nu^{\phi_d\beta_{f*}} \end{aligned}$$

$$\begin{aligned} K_{2,d,f*} &= \left(\theta^{\rho(d)}h\right)^{-t_d} \\ &= \left(g^{\rho(d)a+b} \cdot \prod_{i\in[t]} g^{xz/b_i} \cdot \prod_{i\in[t]} \left(g^{y/b_i^2}\right)^{\rho(d)-ATT_i^*}\right)^{\Upsilon} \\ &\quad \cdot \left(\theta^{\rho(d)}h\right)^{-t'_d} \\ &= \Gamma \cdot \Delta \cdot \prod_{i\in[t]} g^{xyz\langle A_d, \overrightarrow{\eta}\rangle/b_i} \\ &\quad \cdot \prod_{(i,j)\in[t,t]} g^{-xyz\langle A_d, \overrightarrow{\eta}\rangle \frac{b_j(\rho(d)-ATT_i^*)}{b_i^2(\rho(d)-ATT_j^*)}} \\ &= \Gamma \cdot \Delta \\ &\quad \cdot \prod_{(i,j)\in[t,t], i\neq j} \left(g^{xyzb_j/b_i^2}\right)^{-\langle A_d, \overrightarrow{\eta}\rangle \frac{\rho(d)-ATT_i^*}{\rho(d)-ATT_j^*}} \end{aligned}$$

where

$$\Upsilon = y\langle A_d, \overrightarrow{\eta}\rangle - \sum_{i\in[t]} \frac{xzb_i\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d) - ATT_i^*}$$

$$\Gamma = (g^y)^{\langle A_d, \overrightarrow{\eta}\rangle(\rho(d)a+b)} \cdot \prod_{i\in[t]} \left(g^{xzb_i}\right)^{-\frac{(\rho(d)a+b)\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d)-ATT_i^*}} \cdot \left(\theta^{\rho(d)}h\right)^{-t'_d}$$

$$\Delta = \prod_{(i,j)\in[t,t]} \left(g^{(xz)^2 b_j/b_i}\right)^{-\frac{\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d)-ATT_j^*}} \cdot \prod_{i\in[t]} \left(g^{y^2/b_i^2}\right)^{\langle A_d, \overrightarrow{\eta}\rangle(\rho(d)-ATT_i^*)}$$

$$K_{3,d,f*} = g^{t_d} = (g^y)^{-\langle A_d, \overrightarrow{\eta}\rangle} \cdot \prod_{i\in[t]} \left(g^{xzb_i}\right)^{\frac{\langle A_d, \overrightarrow{\eta}\rangle}{\rho(d)-ATT_i^*}} \cdot g^{t'_d}$$

We note that $K_{1,d,f*}$, $K_{2,d,f*}$, and $K_{3,d,f*}$ can be computed by selecting suitable elements from the assumption and are properly distributed for each d.

Challenge: The adversary $\mathcal{A}$ submits two same-length messages M_0 and M_1 for challenge. The simulator $\mathcal{B}$ selects $\mu \in \{0,1\}$ at random. It sets $s = z$ and $r_k = b_k$ for every $k \in [t]$. The elements $s, \{r_k\}_{k\in[t]}$ are correct distributed because the parameters $z, \{b_k\}_{k\in[t]}$ from the q-type assumption are chosen out of the adversary's view. Now, $\mathcal{B}$ can provide the ciphertext for $\mathcal{A}$ by computing

$$C = T \cdot M_\mu$$
$$C_0 = g^s = g^z$$
$$C_1 = \prod\nolimits_{f\in\mathbb{F}_E} g^{\beta_f s} = \prod\nolimits_{f\in\mathbb{F}_E} (g^s)^{\beta_f}$$
$$C_{2,k} = g^{r_k} = g^{b_k}$$

$$C_{3,k} = \left(\theta^{ATT_k^*} h\right)^{r_k} \omega^{-s} = g^{b_k\left(aATT_k^*+b\right)} \cdot \prod_{i\in[t]} g^{xzb_k/b_i} \cdot \prod_{i\in[t]} g^{yb_k\left(ATT_k^*-ATT_i^*\right)/b_i^2} \cdot g^{-xz}$$

$$= \left(g^{b_k}\right)^{aATT_k^*+b} \cdot \prod_{i\in[t],i\neq k} g^{xzb_k/b_i} \cdot \prod_{i\in[t],i\neq k} \left(g^{yb_k/b_i^2}\right)^{ATT_k^*-ATT_i^*}$$

Notice that the elements of the challenge ciphertext can be computed by the terms from the q-type assumption. Finally, the ciphertext is sent to the adversary.

Phase 2: $\mathcal{B}$ acts as in Phase 1.

Guess: The adversary outputs its guess μ' on μ. If $\mu' = \mu$, the simulator answers 0 in the q-type game, that is, it declares that $T = e(g,g)^{xyz}$. Otherwise, it answers 1.

Suppose the adversary can break our selective security game with an advantage ε, we now give the advantage with which the simulator can break the q-type assumption.

If $o = 0$, the challenge ciphertext is valid. Hence, the adversary can guess $\mu' = \mu$ with the advantage ε, That is, $Pr[\mu' = \mu | o = 0] = \frac{1}{2} + \varepsilon$. Because the simulator answers $o' = 0$ when $\mu' = \mu$, we have $Pr[o' = o | o = 0] = \frac{1}{2} + \varepsilon$.

If $o = 1$, the adversary obtains no information about μ. Thereby, we have $Pr[\mu' \neq \mu | o = 1] = \frac{1}{2}$. Because $\mathcal{B}$ answers $o' = 1$ when $\mu' \neq \mu$, we have $Pr[o' = o | o = 1] = \frac{1}{2}$.

Finally, the overall advantage of $\mathcal{B}$ in solving the $q-2$ assumption is $\frac{1}{2}Pr[o' = o | o = 0] + \frac{1}{2}Pr[o' = o | o = 1] - \frac{1}{2} = \frac{1}{2}\varepsilon$.

6. CONCLUSION

In this paper, we presented a large universe decentralized KP-ABE system, where no additional restriction is imposed on the set of attributes that will be taken in encryption. In the proposed scheme, there is no CA. Each participant can be an AA by announcing its attribute universe and the public parameters. Each AA issues the private keys to users are that related to their attributes and can join or depart the system without resetting the system. To prevent collusion attacks, all the user's private keys are linked together by his or her GID. The size of the system public parameters is not relevant to the size of attributes universe. It is proportional to the number of AAs. Our scheme supports any monotonic access structure, which can be expressed by an LSSS matrix, and is almost as efficient as the underlying single-authority KP-ABE system. Finally, we prove the selective security of our scheme in the standard model.

ACKNOWLEDGEMENTS

This research is supported by Changjiang Scholars and Innovative Research Team in the University of China under grant no. IRT1078, the Key Program of NSFC, Guangdong Union Foundation of China under grant no. U1135002, major national S&T program of China under grant no. 2011ZX03005-002, the Fundamental Research Funds for the Central Universities of China under grant no. JY10000903001, and the National Natural Science Foundation of China under grant no. 61370078. We thank the sponsors for their support and the reviewers for helpful comments.

REFERENCES

1. Sahai A, Waters B. Fuzzy identity-based encryption, *Advances in Cryptology–EUROCRYPT 2005*, Aarhus, Denmark, 2005; 557–557.

2. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data, *Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, ACM, 89–98.
3. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption, *Security and Privacy, 2007. SP'07. IEEE Symposium on*, IEEE, 2007; 321–334.
4. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. *Automata, Languages and Programming* 2008; **5126**: 579–591.
5. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, *Public Key Cryptography–PKC 2011*, Taormina, Italy, 2011; 53–70.
6. Cheung L, Newport C. Provably secure ciphertext policy abe, *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, ACM, 2007; 456–465.
7. Lewko A, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption, *Advances in Cryptology–EUROCRYPT 2010*, French Riviera, 2010; 62–91.
8. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures, *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, ACM, 2007; 195–203.
9. Hur J, Park C, Hwang SO. Fine-grained user access control in ciphertext-policy attribute-based encryption. *Security and Communication Networks* 2012; **5** (3): 253–261.
10. Chase M. Multi-authority attribute based encryption, *Proceedings of the 4th conference on Theory of cryptography*, Springer, 2007; 515–534.
11. Chase M, Chow S. Improving privacy and security in multi-authority attribute-based encryption, *Proceedings of the 16th ACM conference on Computer and communications security*, Chicago, Illinois, USA, ACM, 2009; 121–130.
12. Lewko A, Waters B. Decentralizing attribute-based encryption, *Advances in Cryptology–EUROCRYPT 2011*, Tallinn, Estonia, 2011; 568–588.
13. Müller S, Katzenbeisser S, Eckert C. Distributed attribute-based encryption, *Information Security and Cryptology–ICISC 2008*, Seoul, Korea, 2009; 20–36.
14. Müller S, Katzenbeisser S, Eckert C. On multi-authority ciphertext-policy attribute-based encryption. *Bulletin of the Korean Mathematical Society* 2009; **46**(4): 803–819.
15. Lewko A, Waters B. Unbounded hibe and attribute-based encryption, *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology,* EUROCRYPT'11, Springer-Verlag, Berlin, Heidelberg, 2011; 547–567.
16. Liu Z, Cao Z, Huang Q, Wong D, Yuen T. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles, *Computer Security–ESORICS 2011*, Leuven, Belgium, 2011; 278–297.
17. Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption, *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, ACM, 2013; 463–474.
18. Lin H, Cao Z, Liang X, Shao J. Secure threshold multi authority attribute based encryption without a central authority, *Progress in Cryptology-INDOCRYPT 2008*, Kharagpur, India, 2008; 426–436.
19. Li J, Huang Q, Chen X, Chow SSM, Wong DS, Xie D. Multi-authority ciphertext-policy attribute-based encryption with accountability, *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, ACM, New York, NY, USA, 2011; 386–390.
20. Boneh D, Franklin M. Identity-based encryption from the weil pairing, *Advances in Cryptology CRYPTO 2001*, Springer, Santa Barbara, California, USA, 2001; 213–229.
21. Beimel A. Secure schemes for secret sharing and key distribution[D]. *PhD thesis*, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
22. Boneh D, Boyen X. Efficient selective-id secure identity-based encryption without random oracles, *Advances in Cryptology-EUROCRYPT 2004*, Springer, Interlaken, Switzerland, 2004; 223–238.
23. Canetti R, Halevi S, Katz J. Chosen-ciphertext security from identity-based encryption, *Advances in Cryptology-EUROCRYPT 2004*, Springer, Interlaken, Switzerland, 2004; 207–222.
24. Lynn B. *The Pairing-based Cryptography (PBC) library*. http://crypto.stanford.edu/pbc/.