Attribute-Based Encryption Without Key Escrow

Xing Zhang[1(✉)], Cancan Jin[2], Zilong Wen[2], Qingni Shen[2], Yuejian Fang[2], and Zhonghai Wu[2]

[1] School of Electronics Engineering and Computer Science, Peking University, Beijing, China
novostary@gmail.com
[2] School of Software and Microelectronics, Peking University, Beijing, China
jincancan1992@126.com, 450275803@qq.com, {qingnishen,fangyj,zhwu}@ss.pku.edu.cn

Abstract. Attribute-Based Encryption (ABE) is a promising cryptographic primitive for fine-grained sharing of encrypted data. However, ABE has a major shortcoming which is called the key escrow problem. Key generation center (KGC) can generate the secret key of a user with arbitrary set of attributes. Even worse, KGC can decrypt ciphertext directly using its master key. This could be a potential intimidation to data security and privacy. In this paper, we propose a novel ciphertext-policy ABE scheme without key escrow. In our construction, we use two authorities, KGC and OAA (outsourced attribute authority). Unless KGC colludes with OAA, neither KGC nor OAA can decrypt the ciphertext independently. Our scheme is proved to be selectively secure in the standard model. We give universal methods for transforming both KP-ABE and CP-ABE with a single authority to solve the problem of key escrow. Our scheme naturally supports outsourcing the decryption of ciphertexts.

Keywords: Cloud storage · Access control · Attribute-based encryption · Key escrow · Outsourcing decryption

1 Introduction

Do you think that your data storing in the online cloud storage are secure? Although cloud storage service providers, such as Dropbox, Google, Microsoft and so on, announce that they provide security mechanisms for protecting their systems, how about cloud storage service providers themselves? It is convenient for us to access our data anytime and anywhere after moving our data to the cloud. We must remain vigilant on the security and privacy of our data, especially sensitive data. It is better to encrypt sensitive data previous to uploading them to the cloud storage. Thus, even if the cloud storage is broken, the privacy of our data will not be leaked. One shortcoming of encrypting data as a whole is that it severely limits the flexibility of users to share their encrypted data at a fine-grained dimension. Assuming a user wants to grant access permission of all documents of a certain project to a project member, he either needs to

© Springer International Publishing Switzerland 2015
Z. Huang et al. (eds.): ICCCS 2015, LNCS 9483, pp. 74–87, 2015.
DOI: 10.1007/978-3-319-27051-7_7

act as an intermediary and decrypt all relevant files for this member or must give this member his secret decryption key. Neither of these options is particularly attractive. Especially, it is tough when the user wants to share different documents with different people.

Sahai and Waters [1] firstly proposed the concept of Attribute-Based Encryption (ABE) to address this issue. ABE is a cryptographic primitive for fine-grained data access control in one-to-many communication. In traditional Identity-Based Encryption (IBE) [2], the ciphertext is computed according to the targeted user's identity, and only that user himself can decrypt the ciphertext. It is one-to-one communication. As a generalization of IBE, ABE introduces an innovative idea of access structure in public key cryptosystem, making the user's secret key or ciphertext generated based on an access structure. Only the user who meets the specified conditions can decrypt the ciphertext.

Nevertheless, ABE has a major shortcoming which is called the key escrow problem. We clarify the problem as two types: (1) Type 1: key generation center (KGC) can generate a user's secret key with arbitrary access structures or set of attributes, (2) Type 2: KGC can decrypt the ciphertext directly utilizing its master key. These could be potential threats to the data confidentiality and privacy in the cloud storage, thereby affecting the extensive application in the cloud storage.

Why do we need to solve the key escrow problem? Isn't KGC trusted? Let's give an example with public key infrastructure (PKI). A PKI is an arrangement that binds public keys with respective users' identities with a certificate authority (CA). There is one point to note that PKI doesn't know users' secret keys, although PKI is trusted. However, users' secret keys are generated by KGC in ABE. Even if KGC is trusted, we still don't want it to decrypt our encrypted data.

Through our research, we give an informal conclusion that **an ABE scheme has the key escrow problem inherently if there is only one authority (KGC) in the scheme**. The secret key of a user is generated by KGC and there isn't user-specific information in the ciphertext. Otherwise, it will be contrary to the goal of ABE which is designed for fine-grained data sharing. Therefore, we pay our attention to how the cooperation between two authorities to solve the key escrow problem.

1.1 Related Work

Sahai and Waters [1] firstly presented the notion of Attribute-Based Encryption (ABE). Then, ABE comes into two flavors, key-policy ABE (KP-ABE) [3–6] and ciphertext-policy ABE (CP-ABE) [6–9]. In KP-ABE, ciphertexts are associated with sets of attributes and users' secret keys are associated with access structures. In CP-ABE, the situation is reversed, users' secret keys are labeled by attributes and ciphertexts are associated with access structures.

Hur [10, 11] solved the key escrow problem by proposing a secure two-party computation protocol. The original KGC is divided into two parts: KGC and the data storing center. The secure 2PC protocol ensures that neither of them could generate the key all alone. The KGC is accountable for authenticating the user and issues the secret

key to him/her. The drawback of this approach is that it doesn't have universality and it is proved in the random oracle model. Zhang et al. [12] proposed a solution to solve key escrow problem. Zhang et al. introduced another secret key x that KGC does not know. This has some taste of our proposed scheme. However, since the user can acquire x, if the user colludes with KGC, KGC can decrypt any ciphertext. And Zhang et al. just applied this idea for FIBE.

Wang et al. [13] achieved authority accountability by combining Libert and Vergnaud's IBE scheme [14] and KP-ABE [3]. As the user's secret key contains the secret information that KGC does not know, if KGC forges secret keys in accordance with the user's identity, we can fine whether KGC or the user is dishonest according the key family number. However, KGC can still decrypt the ciphertext directly using its master key.

1.2 Our Contributions

The main contributions of our work can be summarized as follows.

(1) We propose a scheme for solving the key escrow problem.
(2) We prove our scheme to be selectively secure in the standard model.
(3) We use two authorities, KGC and OAA (outsourced attribute authority) in our scheme. Our scheme can resist collusion attack from curious KGC or OAA, and even dishonest users colluding with KGC or OAA.
(4) We give universal methods for transforming both KP-ABE and CP-ABE with a single authority to remove the problem of key escrow.
(5) In extensions, we show that we also propose a more practical ABE scheme with outsourcing decryption.

Table 1 shows the comparisons with other related works.

1.3 Our Main Ideas

We will construct our scheme based on CP-ABE of Rouselakis and Waters [6]. There are two challenges for proposing a scheme without key escrow problem. One is how to fragment an authority into two different authorities. We must ensure that any one authority cannot decrypt the ciphertext or generate users' secret keys independently. Moreover, a protocol is necessary for the two authorities to communicate with each other to generate secret keys of users. The other is whether a universal transformation method can remove the key escrow problem from all single authority ABE schemes.

To address the first challenge, a natural idea is to make different authorities have different master keys and perform the same procedure of Key Generation. The user learning both can combine them back into the format of secret key in a single authority ABE. However, the user needs to perform additional calculations to get the final secret key in this trivial idea. If the size of a user's secret key is large, it is inefficient for a user to calculate his/her secret key. Therefore, different authorities cannot perform the same

Table 1. Comparisons with other related works

Scheme	Without Key Escrow		Security Model	Universality
	Type 1	Type 2		
[11]	✓	✓	random oracle model	×
[13]	✓	×	standard model	×
Ours	✓	✓	standard model	✓

procedure of Key Generation trivially. The outsourcing decryption of ABE scheme in Green et al. [15] gives us a hint. In their scheme, decryption step is divided into two stages. Two stages means two decryption keys. Thus, can we use two authorities to generate the two decryption keys independently? We answer this question in the affirmative. However, their scheme still exists the key escrow problem.

To address the second challenge, every scheme has "α" in its master key. Changing "α" can affect the exponent of the user's secret key. The core idea is to provide a method that every authority has part of "α" and neither of them can recover "α" independently. Notice that some schemes [3, 8] use y other than α and y is equivalent to α.

1.4 Organization

The rest of this paper is arranged as follows. Section 2 introduces some cryptographic background information. Section 3 describes the formal definition of CP-ABE without key escrow (WoKE-CP-ABE) and its security model. In Sect. 4, we propose the construction of our WoKE-CP-ABE scheme. In Sect. 5, we analyze the security of our proposed scheme and compare our scheme with multi-authority attribute-based encryption. In Sect. 6, we discuss some extensions. Finally, we conclude this paper.

2 Background

2.1 Access Structure

Definition 2.1 Access Structure [16]. Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall \mathrm{B}, \mathrm{C}$: if $\mathrm{B} \in \mathbb{A}$ and $\mathrm{B} \subseteq \mathrm{C}$ then $\mathrm{C} \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our context, the role of the parties is taken by the attributes. Thus, the access structure $\mathbb{A}$ will contain the authorized sets of attributes. From now on, we focus on monotone access structures.

Definition 2.2 Linear Secret Sharing Schemes (LSSS) [16]. Let $\mathcal{K}$ be a finite field, and $\prod$ be a secret sharing scheme with domain of secrets $S \in \mathcal{K}$ realizing an access structure $\mathcal{A}$. We say that $\prod$ is a linear secret sharing scheme over $\mathcal{K}$ if:

1. The piece of each party is a vector over $\mathcal{K}$. That is, for every i there exists a constant d_i such that the piece of P_i is taken from $\mathcal{K}^{d_i}$. We denote by $\prod_{i,j}(s, r)$ the j-th coordinate in the piece of P_i (where $s \in S$ is a secret and $r \in R$ is the dealer's random input).
2. For every authorized set, the reconstruction function of the secret from the pieces is linear. That is, for every $G \in \mathcal{A}$ there exist constants $\{\alpha_{i,j} : P_i \in G, 1 \leq j \leq d_i\}$, such that for every secret $s \in S$ and every choice of random inputs $r \in R$,

$$s = \sum_{P_i \in G} \sum_{1 \leq j \leq d_i} \alpha_{i,j} \cdot \prod_{i,j}(s, r)$$

where the constants and the arithmetic are over the field $\mathcal{K}$.

The total size of the pieces in the scheme is defined as $d \triangleq \sum_{i=1}^{n} d_i$.

2.2 Bilinear Map

Definition 2.3 Bilinear Map. Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_0$ and e be a bilinear map, $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. The bilinear map e has the following properties:

Bilinearity: for all $u, v \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

Non-degeneracy: $e(g, g) \neq 1$.

Computable: there exists an efficient algorithm for the bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$.

Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.3 Assumption

We state our complexity assumption below.

Definition 2.4 *q*-type Assumption. Initially the challenger calls the group generation algorithm with input the security parameter, picks a random group element $g \in \mathbb{G}_0$, and $q + 2$ random exponents $a, s, b_1, b_2, \ldots, b_q \in \mathbb{Z}_p$. Then he sends to the adversary the group description $(p, \mathbb{G}_0, \mathbb{G}_1, e)$ and all of the following terms:

$$
\begin{aligned}
& g, g^{s} \\
& g^{a^{i}}, g^{b_j}, g^{sb_j}, g^{a^{i}b_j}, g^{a^{i}b_j^2} \qquad \forall (i,j) \in [q,q] \\
& g^{a^{i}b_j/b_{j'}^2} \qquad \forall \left(i,j,j^{'}\right) \in [2q,q,q] with j \neq j^{'} \\
& g^{a^{i}/b_j} \qquad \forall (i,j) \in [2q,q] with i \neq q+1 \\
& g^{sa^{i}b_j/b_{j'}}, g^{sa^{i}b_j/b_j^2} \qquad \forall \left(i,j,j^{'}\right) \in [q,q,q] with j \neq j^{'}
\end{aligned}
$$

There is no probabilistic polynomial-time (PPT) adversary can distinguish $e(g,g)^{sa^{q+1}} \in \mathbb{G}_1$ from an element which is randomly chosen from $\mathbb{G}_1$.

3 CP-ABE Without Key Escrow

3.1 Definition

A WoKE-CP-ABE consists of five algorithms.

KGC-Setup $(1^{\lambda}) \rightarrow (\mathbf{PK}_{KGC}, \mathbf{MK}_{KGC})$ This is a randomized algorithm that takes a security parameter $\lambda \in \mathbb{N}$ as input. It outputs the public parameters $\mathrm{PK}_{\mathrm{KGC}}$ and master key $\mathrm{MK}_{\mathrm{KGC}}$.

OAA-Setup $(\mathbf{PK}_{KGC}) \rightarrow (\mathbf{PK}_{OAA}, \mathbf{MK}_{OAA})$ This is a randomized algorithm that takes $\mathrm{PK}_{\mathrm{KGC}}$ as input. It outputs the public parameters $\mathrm{PK}_{\mathrm{OAA}}$ and master key $\mathrm{MK}_{\mathrm{OAA}}$. The system's public parameters PK can be viewed as $\mathrm{PK}_{\mathrm{KGC}} \cup \mathrm{PK}_{\mathrm{OAA}}$.

Key Generation This is a key issuing protocol. In this protocol, the KGC and OAA generate the user's secret key SK with a set of attributes $\mathcal{S}$ collaboratively.

Encryption $(\mathbf{PK}, \mathbf{M}, \mathcal{T}) \rightarrow \mathbf{CT}$ This is a randomized algorithm that takes as input the public parameters PK, a plaintext message M, and an access structure $\mathcal{T}$. It outputs the ciphertext CT.

Decryption $(\mathbf{CT}, \mathbf{SK}, \mathbf{PK}) \rightarrow \mathbf{M}$ This algorithm takes as input the ciphertext CT that is encrypted under an access structure $\mathcal{T}$, the decryption key SK for a set of attributes $\mathcal{S}$ and the public parameters PK. It outputs the message M if $\mathcal{T}(\mathcal{S}) = 1$.

3.2 Selective Security Model for WoKE-CP-ABE

We define a game for proving the selective security of WoKE-CP-ABE under the chosen plaintext attack.

Init The adversary $\mathcal{A}$ declares the challenge access structure $\mathcal{T}^*$ that he wishes to challenge.

Setup In this stage, the challenger $\mathcal{B}$ simulates **KGC-Setup** and **OAA-Setup** to give the public parameters PK to the adversary $\mathcal{A}$.

Phase 1 The adversary $\mathcal{A}$ issues queries for secret keys for many sets of attributes $\mathcal{S}_i$, where $\mathcal{T}^*(\mathcal{S}_i) = 0$ for all i. The challenger $\mathcal{B}$ calls Key Generation and sends SK_i to the adversary $\mathcal{A}$.

Challenge The adversary $\mathcal{A}$ submits two equal length messages M_0 and M_1. The challenger $\mathcal{B}$ flips a random coin b, and encrypts M_b with $\mathcal{T}^*$. Then $\mathcal{B}$ passes the ciphertext to the adversary $\mathcal{A}$.

Phase 2 Phase 1 is repeated.

Guess The adversary $\mathcal{A}$ outputs a guess b' of b.

The advantage of an adversary $\mathcal{A}$ in this game is defined as $|\Pr[b' = b] - 1/2|$.

Definition 3.1. A ciphertext-policy attribute-based encryption scheme without key escrow is selectively secure if all PPT adversaries have at most negligible advantage in λ in the above security game.

3.3 Key Escrow Model for WoKE-CP-ABE

There are also other four types of adversaries. The adversary above (named **Type-I Adversary**) focuses on ABE scheme, and the below focus on the problem of key escrow.

Type-II Adversary. It is defined as a curious KGC. Such an adversary owns part of master key of the system and tries to extract useful information from ciphertext. However, this type adversary is restricted that he cannot collude with any user or OAA.

Type-III Adversary. It is defined as a curious OAA. This adversary is similar to type-II adversary, except the part of master key he owns. Notice that the restriction is that he cannot collude with any user or KGC.

Type-IV Adversary. It is defined as dishonest users colluding with KGC. Such an adversary owns KGC's master key and is allowed to ask for all secret keys SK of dishonest users. The goal of this adversary is to obtain useful information from ciphertext not intended for him. Notice that Type-IV adversary cannot collude with OAA.

Type-V Adversary. It is defined as dishonest users colluding with OAA. This adversary is similar to type-IV adversary except the users can collude with OAA instead of KGC. Notice that Type-V adversary cannot collude with KGC.

As we know, we suggest these four adversaries because we must show the construction is key escrow resistant. We must take type-II and type-III adversary into consideration because our scheme must prevent any key generation authority from attacking the scheme. Then type-IV adversary and type-V adversary are some kind of more "powerful" adversary, they want to gain some information about the adverse organization secret key, thus decrypting some message not intended for them.

Notice that a reasonable assumption is that KGC cannot collude with OAA. Otherwise, our scheme can be viewed as a scheme with only a single authority and this "authority" can decrypt any ciphertext according to our analysis in Sect. 1.

4 Our Construction

Let $\mathbb{G}_0$ be a bilinear group of prime order p, and let g be a generator of $\mathbb{G}_0$. In addition, let $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ denote the bilinear map. A security parameter λ will determine the size of the groups. For the moment we assume that attributes are elements in $\mathbb{Z}_p^*$.

Nevertheless, attributes can be any meaningful unique strings using a collision resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$.

Our construction follows.

KGC-Setup $(1^{\lambda}) \rightarrow (\boldsymbol{PK_{KGC}}, \boldsymbol{MK_{KGC}})$ The algorithm calls the group generator algorithm $\mathcal{G}(1^{\lambda})$ and gets the descriptions of the groups and the bilinear map $D = (p, \mathbb{G}_0, \mathbb{G}_1, e)$. Then choose the random terms $g, u, h, w, v \in \mathbb{G}_0$ and $\alpha^{'} \in \mathbb{Z}_p$. The published public parameters PK_{KGC} are

$$\left(D, g, u, h, w, v, e(g,g)^{\alpha^{'}}\right).$$

The master key MK_{KGC} is $\alpha^{'}$.

OAA-Setup $(\boldsymbol{PK_{KGC}}) \rightarrow (\boldsymbol{PK_{OAA}}, \boldsymbol{MK_{OAA}})$ Choose μ uniformly at random in $\mathbb{Z}_p$. We can view $\alpha^{'}$ as α/μ. The published public parameters PK_{OAA} is

$$(e(g,g)^{\alpha^{'}})^{\mu} = e(g,g)^{\alpha}.$$

The master key MK_{OAA} is μ.

Then, the system's public parameters PK can be viewed as

$$PK = (D, g, u, h, w, v, e(g,g)^{\alpha}).$$

Key Generation KGC and OAA are involved in the user's key issuing protocol. In the protocol, KGC needs to communicate OAA to generate the user's secret key. The key issuing protocol consists of the following steps:

1. Firstly, KGC and OAA authenticate a user U with set of attributes $\mathcal{S} = \{A_1, A_2, \ldots, A_k\} \subseteq \mathbb{Z}_p$ independently.
2. KGC selects a random exponent $\theta \in_R \mathbb{Z}_p^*$ and sends it to U. θ is used to prevent OAA from obtaining U's complete secret key.
3. KGC picks $r^{'}, r_1^{'}, r_2^{'}, \ldots, r_k^{'} \in_R \mathbb{Z}_p^*$ and computes

$$\mathcal{S}, K_0^{'} = g^{\alpha^{'}/\theta} w^{r^{'}/\theta}, w^{1/\theta}, K_1^{'} = g^{r^{'}},$$

$$\{K_{i,2}^{'} = g^{r_i^{'}}, K_{i,3}^{'} = (u^{A_i} h)^{r_i^{'}} v^{-r^{'}}\}_{i \in [k]}.$$

Then send it to OAA.

4. OAA chooses $r^{''}, r_1^{''}, r_2^{''}, \ldots, r_k^{''} \in_R \mathbb{Z}_p^*$ and computes

$$\mathcal{S}, K_0^{''} = (K_0^{'})^{\mu} \cdot (w^{1/\theta})^{r^{''}}, K_1^{''} = (K_1^{'})^{\mu} \cdot g^{r^{''}},$$

$$\{K_{i,2}^{''} = (K_{i,2}^{'})^{\mu} \cdot g^{r_i^{''}}, K_{i,3}^{''} = (K_{i,3}^{'})^{\mu} \cdot (u^{A_i} h)^{r_i^{''}} v^{-r^{''}}\}_{i \in [k]}.$$

Then send it to the user.

Notice that the role of $r^{''}, r_1^{''}, r_2^{''}, \ldots, r_k^{''}$ is to randomize the secret key. Otherwise, if a dishonest user colluded with KGC, KGC can compute $((g^{r^{'}})^{\mu})^{1/r^{'}} = g^{\mu}$ and use it to decrypt any ciphertext by calculating $me(g,g)^{\alpha s}/(e(g^s, g^{\mu}))^{\alpha^{'}} = m$.

5. The user obtains his/her secret key as

$$SK = (\mathcal{S}, \theta, K_0 = K_0^{''} = g^{\alpha^{'}\mu/\theta} w^{(r^{'}\mu + r^{''})/\theta} = g^{\alpha/\theta} w^{r/\theta}, K_1 = K_1^{''} = g^{r^{'}\mu + r^{''}} = g^{r},$$

$$\{K_{i,2} = K_{i,2}^{''} = g^{r_i^{'}\mu + r_i^{''}} = g^{r_i}, K_{i,3} = K_{i,3}^{''} = \left(u^{A_i} h\right)^{r_i^{'}\mu + r_i^{''}} v^{-\left(r^{'}\mu + r^{''}\right)} = (u^{A_i} h)^{r_i} v^{-r}\}_{i \in [k]}).$$

We implicitly set $r = r^{'}\mu + r^{''}, \{r_i = r_i^{'}\mu + r_i^{''}\}_{i \in [k]}$.

Encryption $(\boldsymbol{PK}, \boldsymbol{m}, (\boldsymbol{M}, \boldsymbol{\rho})) \rightarrow \boldsymbol{CT}$ To encrypt a message $m \in \mathbb{G}_1$ under an access structure encoded in an LSSS policy (M, ρ). Let the dimensions of M be $l \times n$. Each row of M will be labeled by an attribute and $\rho(i)$ denotes the label of i^{th} row $\vec{M}_i$. Choose a random vector $\vec{z} = (s, z_2, \ldots, z_n)^T$ from $\mathbb{Z}_p^n$, s is the random secret to be shared among the shares. The vector of the shares is $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_l)^T = M\vec{z}$. It then chooses l random value $t_1, t_2, \ldots, t_l \in \mathbb{Z}_p$ and publish the ciphertext as:

$$CT = ((M, \rho), me(g,g)^{\alpha s}, C_0 = g^s,$$

$$\{C_{i,1} = w^{\lambda_i} v^{t_i}, C_{i,2} = \left(u^{\rho(i)} h\right)^{-t_i}, C_{i,3} = g^{t_i}\}_{i \in [l]}).$$

Decryption $(\boldsymbol{CT}, \boldsymbol{SK}, \boldsymbol{PK}) \rightarrow \boldsymbol{m}$ To decrypt the ciphertext CT with the decryption key SK, proceed as follows. Suppose that $\mathcal{S}$ satisfies the access structure and let $I = \{i : \rho(i) \in \mathcal{S}\}$. Since the set of attributes satisfy the access structure, there exist coefficients $\omega_i \in \mathbb{Z}_p$ such that $\sum_{\rho(i) \in I} \omega_i \cdot \vec{M}_i = (1, 0, \ldots, 0)$. Then we have that $\sum_{\rho(i) \in I} \omega_i \lambda_i = s$. Now it calculates

$$\frac{me(g,g)^{\alpha s} \prod_{i \in I} (e\left(C_{i,1}, K_1\right) e(C_{i,2}, K_{i,2}) e(C_{i,3}, K_{i,3}))^{\omega_i}}{(e(C_0, K_0))^{\theta}}$$

$$= \frac{me(g,g)^{\alpha s} \prod_{i \in I} (e\left(w^{\lambda_i} v^{t_i}, g^r\right) e(\left(u^{\rho(i)} h\right)^{-t_i}, g^{r_i}) e(g^{t_i}, (u^{A_i} h)^{r_i} v^{-r}))^{\omega_i}}{(e(g^s, g^{\alpha/\theta} w^{r/\theta}))^{\theta}}$$

$$= \frac{me(g,g)^{\alpha s} e(w,g)^{rs}}{e(g^s, g^{\alpha} w^r)} = m.$$

5 Analysis of Our Proposed Scheme

5.1 Selective Security Proof

In the selective security proof, we will reduce the selective security of our CP-ABE scheme to that of Rouselakis and Waters' [6] which is proved selectively secure under the q-type assumption in Sect. 2.3.

Due to space limited, we have to omit the full proof process.

5.2 Security Analysis for Problem of Key Escrow

Type-II adversary. Type-II adversary is defined as a curious KGC and restricted that he cannot collude with any user or OAA. The adversary needs to recover μ or s to decrypt ciphertext $me(g,g)^{\alpha s}$.

$$(e(g^{\alpha'}, g^{s}))^{\mu} \rightarrow e(g,g)^{\alpha s} \rightarrow m \text{ by using } \mu,$$

$$(e(g,g)^{\alpha})^{s} \rightarrow m \text{ by using } s.$$

However, it is related with discrete logarithm to compute μ or s. Since computing discrete logarithm is believed to be difficult, our scheme can resist the attack from Type-II adversary. □

Type-III adversary. This adversary is similar to type-II adversary. The adversary needs to recover α' or s to decrypt ciphertext $me(g,g)^{\alpha s}$. Since computing discrete logarithm is believed to be difficult, our scheme can resist the attack from Type-III adversary. □

Type-IV adversary. Type-IV adversary is defined as dishonest users colluding with KGC. Although this adversary can request some users' secret keys, he cannot obtain more information about μ than Type-II adversary as the users' secret keys are randomized by OAA. This adversary also needs to recover μ or s to decrypt ciphertext. Since computing discrete logarithm is believed to be difficult, our scheme can resist the attack from Type-IV adversary. □

Type-V adversary. Type-V adversary is defined as dishonest users colluding with OAA. Obviously, this adversary has less power than Type-IV adversary. The adversary needs to recover α', s or r' to decrypt ciphertext.

$$(g^{\alpha'/\theta} w^{r'/\theta})^{\theta\mu} \rightarrow g^{\alpha} w^{r'\mu}.$$

If this adversary knows r', he can calculate any set of attributes by using θ from dishonest users. r' is related with discrete logarithm problem. So our scheme can resist the attack from Type-V adversary. □

5.3 Comparing with Multi-authority Attribute-Based Encryption

In multi-authority ABE [17–20], different authorities manage different sets of attributes. Chase [17] proposed a multi-authority ABE scheme using the concepts of a trusted central authority (CA) and global identifiers (GID). However, the CA has the power to decrypt any ciphertext. Chase and Chow [18] proposed a multi-authority ABE scheme without a trusted central authority (CA). However, all attribute authorities (AAs) must communicate among each other in a secure two party key exchange protocol and each authority also needs to give non-interactive proofs of knowledge of some parameters. If the number of colluded AAs is less than $N-1$ (N is the total number of AAs in the system), the colluded parties cannot decrypt ciphertexts which policies do not satisfy these parties' attributes. Nevertheless, an AA has absolute control over attributes belonging to itself. If an AA colludes with a dishonest user, it is equivalent that they have all the attributes keys belonging to this AA. They can decrypt more ciphertexts than this user. Our scheme can resist collusion attack from curious KGC or OAA, and even dishonest users colluding with KGC or OAA as both KGC and OAA involve in the generation of every attribute secret key. Additionally, if we apply multi-authority ABE to the scenario that users store their daily data into the public cloud, it is a problem that how to divide the sets of attributes. In a similar scenario, it will be more reasonable to adopt our scheme which enhances a single authority ABE without key escrow. When there are a lot of users, we can manage them by using a hierarchical approach [21].

6 Extensions

6.1 Universality

Our method for removing key escrow can be applicable to other ABE schemes. It is easy to transform a single authority of an ABE scheme to KGC and OAA. At setup stage, KGC performs the same Setup as the original scheme. OAA performs exponent operation on α-related part $e(g,g)^{\alpha}$.

Now we will mainly focus on key generation. We will analyze universality for CP-ABE and KP-ABE respectively.

For CP-ABE, we will describe the transformation method by analyzing our proposed scheme. The main difference between secret key of our scheme and Rouselakis and Waters' [6] is the generation of K_0. In our scheme,

$$K_0^{'} = g^{\alpha^{'}/\theta} w^{r^{'}/\theta}, w^{1/\theta}, \theta \rightarrow (K_0^{'})^{\mu} \cdot \left(w^{1/\theta}\right)^{r^{''}} = g^{\alpha^{'}\mu/\theta} w^{(r^{'}\mu + r^{''})/\theta}$$
$$= g^{\alpha/\theta} w^{r/\theta}, \theta.$$

In Rouselakis and Waters [6], $K_0 = g^{\alpha} w^{r}$. When key generating, we use two more parts, $w^{1/\theta}$ and θ. We can call θ as the key-escrow-free part and $w^{1/\theta}$ as the affiliated part for randomization of secret key. Notice that OAA also needs to randomize the

secret key. The reason we have analyzed in Sect. 4. By using these two parts, we have already been able to construct a scheme without key escrow.

For KP-ABE, there is a little different from CP-ABE. Let us look at an example. For KP-ABE in Rouselakis and Waters' [6], KGC also needs to generate a key-escrow-free part θ and sends it to the user. Then KGC uses the algorithm of Key Generation in [6], the only difference is replacing α with $\alpha^{'}/\theta$. KGC sends it to OAA. OAA performs exponent operation with μ and randomization operation similarly with ours. Then OAA sends it to the user. From this example we can see that we only handle on exponent. It doesn't matter the format of the user's secret key in the original scheme.

The algorithm of Encryption is identical and θ is used in the algorithm of Decryption. As it is apparent, we will not analyze it any more.

6.2 A More Practical ABE Scheme with Outsourcing Decryption

One of the main performance limitations of ABE is that the size of the ciphertext and the time required to decrypt it grow with the complexity of the access structure. As computing power is limited in the mobile devices, it is unacceptable for a long decryption time. Green et al. [15] proposed a method for outsourcing the decryption of ABE ciphertexts to save the user's bandwidth and decryption time in mobile scenarios. The user has a transformation key and an ElGamal-style key. The transformation key can be given to a proxy to translate any ABE ciphertext satisfied by that user's attributes into an ElGamal-style ciphertext without revealing ciphertext's content to that proxy. Then this user can use the ElGamal-style key to decrypt that ElGamal-style ciphertext efficiently. Green et al. [15] need to perform exponential operations on every element in $\mathbb{G}_0$ of the user's secret key by using $1/z$. Although this procedure is performed by KGC in that scheme, we note that it can also be performed by the user. However, if a user wants to change his/her transformation key after using that key too much times, he/she needs to perform exponential operations on every element of his/her secret key. We are surprised to find that our scheme naturally supports outsourcing the decryption of ciphertexts. θ is already our ElGamal-style key and other parts of secret key are the transformation key. If a user wants to change θ to $\theta^{'}$, he/she only needs to calculate $K_0^{\theta/\theta^{'}} = \left(g^{\alpha/\theta} w^{r/\theta}\right)^{\theta/\theta^{'}} = g^{\alpha/\theta^{'}} w^{r/\theta^{'}}$ and updates new K_0 to the proxy. It's very practical and saves the transmission bandwidth for update.

7 Conclusion

Key escrow is quite a challenging issue in ABE. We formalize the concept of ciphertext policy attribute-based encryption without key escrow (WoKE-CP-ABE) and propose a scheme for solving the key escrow problem. In our construction, we use two authorities, KGC and OAA (outsourced attribute authority) which communicate with each other to issue secret keys for users. Unless KGC colludes with OAA, neither KGC nor OAA can decrypt the ciphertext independently. Our scheme is proved to be selectively

secure in the standard model. We give universal methods for transforming both KP-ABE and CP-ABE with a single authority to solve the problem of key escrow. In addition, our scheme naturally supports outsourcing the decryption of ciphertexts. As KGC's behavior is restricted in ABE with a single authority, it will drive people to store more sensitive data into cloud and promote the application of ABE in a wider range.

Acknowledgments. This work is supported by the National High Technology Research and Development Program ("863" Program) of China under Grant No. 2015AA016009, the National Natural Science Foundation of China under Grant No. 61232005, and the Science and Technology Program of Shen Zhen, China under Grant No. JSGG2014051 6162852628.

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
2. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
4. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
5. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)
6. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM Conference on Computer and Communications Security, pp. 463–474 (2013)
7. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
8. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
9. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
10. Hur, J., Koo, D., Hwang, S.O., Kang, K.: Removing escrow from ciphertext policy attribute-based encryption. Comput. Math Appl. **65**(9), 1310–1317 (2013)
11. Hur, J.: Improving security and efficiency in attribute-based data sharing. IEEE Trans. Knowl. Data Eng. **25**(10), 2271–2282 (2013)
12. Zhang, G., Liu, L., Liu, Y.: An attribute-based encryption scheme secure against malicious KGC. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1376–1380 (2012)
13. Wang, Y., Chen, K., Long, Y., Liu, Z.: Accountable authority key policy attribute-based encryption. Sci. China Inf. Sci. **55**(7), 1631–1638 (2012)

14. Libert, B., Vergnaud, D.: Towards black-box accountable authority IBE with short ciphertexts and private keys. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 235–255. Springer, Heidelberg (2009)
15. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: USENIX Security Symposium (2011)
16. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
17. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
18. Chase, M., Chow, S.S.: Improving privacy and security in multi-authority attribute-based encryption. In: ACM Conference on Computer and Communications Security, pp. 121–130 (2009)
19. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011)
20. Liu, Z., Cao, Z., Huang, Q., Wong, D.S., Yuen, T.H.: Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 278–297. Springer, Heidelberg (2011)
21. Wang, G., Liu, Q., Wu, J.: Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 735–737 (2010)