ATLANTIS PRESS

Advances in Engineering Research, volume 86

2nd International Conference on Electrical, Automation and Mechanical Engineering (EAME 2017)

Large Universe Ciphertext-Policy Attribute-Based Encryption with Efficient Revocation

Lei Sun, Shuaili Wang*, Zuohui Li and Guangbo Wang

Zhengzhou Information Science and Technology Institute, Zhengzhou 450004, China

*Corresponding author

***Abstract*—In this paper, we propose a large universe ciphertext-policy attribute based encryption (ABE) scheme with efficient revocation. To achieve the revocation, we divide the master key into two parts: delegation key and secret key, which are sent to the cloud provider and user separately. Note that, our scheme is proved selectively secure in the standard model under q-type assumption. Finally, we give the concrete analysis associated with our scheme including security requirements, functionality and performance.**

Keywords-ciphertext-policy attribute-based encryption; large universe; attribute level user revocatio

I. INTRODUCTION

Attribute Based Encryption (ABE), which was first proposed by Sahai et al. [1] in 2005, has been developed as a cryptographic primitive to achieve fine-grained access control for encrypted data. In ABE, the user can specify the access control policy for encrypted data over a set of attributes, and each user will be issued the corresponding private key from an authority center that reflects the attributes they have. A user will be able to decrypt the ciphertext only if the attributes corresponding to their private satisfy the access control policy ascribed to the ciphertext. Since then, several ABE constructions have been proposed. Goyal et al. proposed an expressive Key-Policy ABE(KP-ABE) scheme, and formalized the notion of Ciphertext-Policy ABE(CP-ABE) [2] followed by other CP-ABE schemes [3-5] and KP-ABE schemes [6-8]. However, a limitation of these ABE schemes is that the system parameters must be chosen at the setup phase, which cannot offer complex flexibility. Lewko et al. first solved this problem by introducing a classification of ABE schemes: small universe and large universe [9]. In the small universe schemes, the size of attribute size is polynomial to the system parameter and must be set at the initial phase. More importantly, the public parameters increase linearly with the size of the universe. In the large universe ABE schemes, the attribute universe can be arbitrarily large and the public parameters can keep constant. Afterwards, Rouselakis et al. proposed two large universe ABE schemes (one CP-ABE and one KP-ABE) on prime order groups and proved secure under q-type assumptions in the standard model [10]. However, it does not involve the attribute revocation which is critical to cloud storage environment. Since each attribute can be shared by multiple users, so the scheme devised must ensure that the revocation will not affect other users in the attribute group.

The fine-grained attribute revocation has got extensive application in many practical ABE schemes. Bethencourt et al. first solved this issue by adding an expiration time to the users' attributes [3] which disabled a user's secret key at a designated time. Next, Yu et al. proposed the first formal key revocation scheme [11], where a proxy performs the re-encryption. The key revocation algorithm relies on an asymmetric algorithm, which limits its efficiency for large data. So it is only suitable for re-encrypting short data. Xie et al. proposed an efficient attribute revocation scheme [12] which used the encryption key tree for each user. The new generated key is used to re-encrypt all the ciphertext by the CSP, which may cause high computation cost. Yang et al. aslo proposed an attribute revocation scheme [13] by using the authority center to update ciphertext and generate new keys including secret new and updating key. However, this scheme poses high computation cost on the authority center and brings high communication cost between the authority center and users. Additionally, all these schemes are designed for ABE with small universe, which limits its extensive application.

In this paper, we propose an efficient and revocable CP-ABE scheme that combines proxy re-encryption method to achieve the attribute revocation. In this scheme, we achieve the revocation with the help of CSP, which offloads most of revocation operations for the authority. The keys are divided into two forms: the secret key for user and the delegation key for CSP, and the delegation key is used to re-encrypt the ciphertext. Only the users whose attributes satisfy the access structure, can update the secret keys and further decrypt the re-encrypted ciphertext.

II. PRELIMINARIES

In this section, we shortly introduce some background information for this paper, including bilinear maps, access structure and q-type assumption.

A. Bilinear Maps

Definition 1. (Bilinear Map). Let ψ be a group parameters generation algorithm which takes as input the security parameter λ and outputs the group parameters $(p,\mathbb{G},\mathbb{G}_T,e)$ .In these group parameters, p denotes a big prime whose size is determined by the security parameter λ , $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with order p , $e:G\times G\rightarrow G_T$ is a bilinear map satisfying the following properties:

(1) Bilinearity: $\forall u,v \in G, a,b \in Z_p$, we have $e(u^a, v^b) = e(u,v)^{ab}$.

(2) Non-degeneracy: $\exists g \in G$ satisfying that $e(g,g)$ has order p in $\mathbb{G}_T$.

B. Access Structure and Linear Secret-Sharing Schemes

Definition 2. (Access Structure [14]) Let $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ be a set of parties. A collection $D \subset 2^P$ is said to be monotone if for all B, C if $B \in D$ and $B \subseteq C$, then $C \in D$ holds. An access structure is a collection $D \subset 2^P \setminus \{0\}$. And the sets in D are called the authorized sets, and the sets are not in D are called the unauthorized sets.

C. q-type Assumption

The assumption is proved via the following game between a challenger and an attacker which is demonstrated as follows:

$$g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i/b_j^2} \quad \forall (i,j) \in [q,q]$$
$$g^{a^i b_j / b_{j'}^2} \quad \forall (i,j,j') \in [2q,q,q] \ \ with \ \ j \neq j'$$
$$g^{a^i/b_j} \quad \forall (i,j) \in [2q,q] \ \ with \ \ i \neq q+1$$
$$g^{sa^i b_j / b_{j'}}, g^{sa^i b_j / b_{j'}^2} \quad \forall (i,j,j') \in [2q,q,q] \ \ with \ \ j \neq j'$$

Next, the challenger will flip a random coin $b \in \{0,1\}$, and if $b = 0$, it gives the term $e(g,g)^{sa^{q+1}}$ to the attacker. Otherwise, it randomly chooses a term $R \in G_T$ and gives it to the attacker. At last, the attacker outputs a bit $b' \in \{0,1\}$ as its guess for b .

Definition 3. We say that q-type assumption holds if polynomial algorithm has a non-negligible advantage to break the above security game, where the advantage is defined as $Adv = |\Pr(b' = b) - 1/2|$.

III. Definition and Security Model

A. Construction

In this section, we propose our revocable large universe CP-ABE scheme partially based on Rouselakis et al.' construction [10] as follows:

Setup(1^λ) $\rightarrow$ (PK, MK). The algorithm first runs the group generator $G(1^\lambda)$ to obtain $D = (p, \mathbb{G}, \mathbb{G}_T, e)$ where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order p , and $e: G \times G \rightarrow G_T$ is a bilinear map. The attribute universe is $\mathcal{U} = \mathbb{Z}_p$.

Then the algorithm randomly chooses parameters $g, u, h, w, v \in G$ and $\alpha_1, \alpha_2 \in Z_p$ such that $\alpha_1 + \alpha_2 = \alpha \ mod \ p$. Finally, the public key PK is set as $PK = (D, g, u, h, w, v, e(g,g)^\alpha)$ .The master key MK is set as $MK = (\alpha_1, \alpha_2)$.

KeyGen ($MK, S = \{A_1, A_2, ..., A_k\} \subseteq Z_p$) $\rightarrow$ (SK_1, SK_2). The key generation algorithm firstly randomly chooses $k+1$ exponents $r, r_1, r_2, ..., r_k \in Z_p$. Then it uses one part of the master key α_1 to compute the user's secret key as $K_0 = g^{\alpha_1} w^r, K_1 = g^r$, and for each attribute $i \in [k]$

$$K_{i,2} = g^{r_i}, \ K_{i,3} = (u^{A_i} h)^{r_i} v^{-r} .$$

Therefore, it sets the secret key $SK_1 = (K_0, K_1, \{K_{i,2}, K_{i,3}\}_{i=1}^k)$. Next, this algorithm uses the other part of the master key α_2 to generate the delegation key as $SK_2 = (D_c = g^{\alpha_2})$ for the CSP.

Encrypt ($m, \mathbb{A}$). $\rightarrow$. CT : The encryption algorithm takes as input the plaintext message m and the access structure $\mathbb{A}$ encoded as an LSSS policy with access matrix $M \in Z_p^{l \times n}$ and map function $\rho: [l] \rightarrow Z_p$. Then the algorithm chooses random exponents $t_1, t_2, ..., t_l$ and a random vector $\vec{v} = (s, y_2, ..., y_n) \in Z_p^n$, where s is the secret to be shared. Next, for $i = 1$ to l , the algorithm computes $\lambda_i = M_i \vec{v}$. Finally, the ciphertext is published as $CT = (A, C, C_0, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [l]})$, where $C = m \cdot e(g,g)^{\alpha s}$, $C_0 = g^s$, and for each attribute $i \in [l]$:

$$C_{i,1} = w^{\lambda_i} v^{t_i}, C_{i,2} = (u^{\rho(i)} h)^{-t_i}, C_{i,3} = g^{t_i} .$$

Re-encrypt ($CT, SK_2, RL_{x'}$) $\rightarrow$ RCT : The re-encryption algorithm takes as input the ciphertext CT , the delegation key SK_2 and the revocation list $RL_{x'}$ of attribute x' . When a user's attribute is revoked, the ciphertext should be re-encrypted to prevent the user from continuing to access it without affecting other legitimate users' normal access. We denote ID_i as the identity of user i .

If there is no attribute revoked, namely $RL_{x'} = \Phi$, then the CSP chooses a random $k \in Z_p$ to encrypt the delegation key g^{α_2} . Finally, it re-encrypts the ciphertext CT as follows:

$$C' = C = m \cdot e(g,g)^{\alpha s}, C_0' = C_0 = g^s, C_1' = g^{s/k} ,$$

$$\forall i = 1, 2, ..., l :$$

$$C_{i,1}' = w^{\lambda_i} v^{t_i} v^k, C_{i,2}' = (u^{\rho(i)} h)^{-t_i} (u^{\rho(i)} h)^{-k}, C_{i,3}' = g^{t_i} g^k$$

Therefore, the re-encrypted ciphertext is set as $CT' = (C', C'_0, C'_1, \{C'_{i,1}, C'_{i,2}, C'_{i,3}\}_{i=1}^{l})$. In addition, the re-encryption algorithm will generate the updated delegation key as $SK'_2 = (D'_c = (g^{\alpha_2})^k)$.

If there is an attribute x' revoked from a user ID_j, namely $RL_{x'} \neq \Phi$, then the CSP will chooses a random exponent $v_{x'} \in Z_p$ and encrypt it as $\hat{C}$ under the access structure (M, ρ) for those users $ID_i, i \neq j$ who possess the revoked attribute and have not been revoked.

Then the CSP also chooses a random $k \in Z_p$ to encrypt the delegation key g^{α_2} and re-encrypts the ciphertext CT as follows:

$$C' = C = m \cdot e(g,g)^{\alpha s}, C'_0 = C_0 = g^s, C'_1 = g^{s/k}$$

$$\forall i = 1, 2, ..., l:$$

$$C'_{i,1} = w^{\lambda_i} v^{t_i} v^k, C'_{i,2} = (u^{\rho(i)} h)^{-t_i} (u^{\rho(i)} h)^{-k}$$

$$for\ \rho(i) \neq x: C'_{i,3} = g^{t_i} g^k$$
$$for\ \rho(i) = x: C'_{i,3} = (g^{t_i} g^k)^{1/v_{\rho(i)}}$$

Therefore, the re-encrypted ciphertext is set as $CT' = (\hat{C}, C', C'_0, C'_1, \{C'_{i,1}, C'_{i,2}, C'_{i,3}\}_{i=1}^{l})$. In addition, the re-encryption algorithm also generates the updated delegation key as $SK'_2 = (D'_c = (g^{\alpha_2})^k)$.

Decrypt $(CT', SK_1, SK'_2) \rightarrow m$: The decryption algorithm takes as input the ciphertext CT' for the access structure (M, ρ), the secret key SK_1 for the attributes set S and the updated delegation key SK'_2.

If there is no attribute revoked, then the decryption computes the set of rows in M that provides a share to the attributes in S, namely $I = \{i : \rho(i) \in S\}$. Then it computes the constants $\{w_i \in Z_p\}_{i \in I}$ such that $\sum_{i \in I} w_i M_i = (1, 0, ..., 0)$. Next, it computes:

$$\mathrm{A} = \prod_{i \in I} (e(C'_{i,1}, K_1) e(C'_{i,2}, K_{i,2}) e(C'_{i,3}, K_{i,3}))^{w_i}$$

Then the message m can be revealed by computing $(C' \cdot \mathrm{A}) / (e(C'_1, SK'_2) \cdot e(C'_0, K_0))$.

If there is an attribute x' revoked from a user ID_j. The user ID_i, $i \neq j$ holds the revoked attribute x' but has not been revoked, and his attributes set S satisfies the access structure (M, ρ), then the decryption algorithm decrypts $\hat{C}$ successfully by using the secret key SK_1 to obtain $v_{x'}$ so that it can update the secret key component $K_{i,3}$ as $K'_{i,3} = ((u^{x'} h)^{r_i} v^{-r})^{v_{x'}}$. Otherwise, it cannot get the updated secret key $K_{i,3}$. Then the decryption algorithm computes the set of rows in M that provides a share to the attributes in S, namely $I = \{i : \rho(i) \in S\}$ and also computes the constants $\{w_i \in Z_p\}_{i \in I}$ such that $\sum_{i \in I} w_i M_i = (1, 0, ..., 0)$. Next, it proceeds as follows:

$$for\ \rho(i) \neq x: B_i = (e(C'_{i,1}, K_1) e(C'_{i,2}, K_{i,2}) e(C'_{i,3}, K_{i,3}))^{w_i} = e(g,w)^{r\lambda_i w_i}$$
$$for\ \rho(i) = x: B_i = (e(C'_{i,1}, K_1) e(C'_{i,2}, K_{i,2}) e(C'_{i,3}, K'_{i,3}))^{w_i} = e(g,w)^{r\lambda_i w_i}$$

$$\mathrm{A} = \prod_{i \in I} B_i = e(g,w)^{rs}$$

Then the message m can be revealed by computing $(C' \cdot \mathrm{A}) / (e(C'_1, SK'_2) \cdot e(C'_0, K_0))$.

B. Correctness

In this part, we will validate the correctness of our proposed scheme by the following equations.

If there is no attribute revoked, then we have

$$\begin{aligned}
\mathrm{A} &= \prod_{i \in I} (e(C'_{i,1}, K_1) e(C'_{i,2}, K_{i,2}) e(C'_{i,3}, K_{i,3}))^{w_i} \\
&= \prod_{i \in I} (e(w^{\lambda_i} v^{t_i} v^k, g^r) e((u^{\rho(i)} h)^{-t_i} (u^{\rho(i)} h)^{-k}, g^{r_i}) e(g^{t_i} g^k, (u^{A_i} h)^{r_i} v^{-r}))^{w_i} \\
&= \prod_{i \in I} (e(w,g)^{r\lambda_i} e(v,g)^{r(k+t_i)} e(u^{\rho(i)} h, g)^{r_i(-t_i-k)} e(g, u^{A_i} h)^{r_i(t_i+k)} e(g,v)^{-r(k+t_i)})^{w_i} \\
&= \prod_{i \in I} e(w,g)^{r\lambda_i w_i} \\
&= e(g,w)^{rs}
\end{aligned}$$

If there is an attribute x' revoked from a user ID_j, then we have

$$
\begin{aligned}
&for\ \rho(i) \neq x':\\
&\quad B_i = (e(C'_{i,1}, K_1)e(C'_{i,2}, K_{i,2})e(C'_{i,3}, K_{i,3}))^{w_i}\\
&\quad\quad = (e(w^{\lambda_i} v^{t_i} v^k, g^r)e((u^{\rho(i)}h)^{-t_i}(u^{\rho(i)}h)^{-k}, g^{r_i})e(g^{t_i} g^k, (u^{A_i} h)^{r_i} v^{-r}))^{w_i}\\
&\quad\quad = (e(w,g)^{r\lambda_i} e(v,g)^{r(k+t_i)} e(u^{\rho(i)}h, g)^{r_i(-t_i-k)} e(g, u^{A_i} h)^{r_i(t_i+k)} e(g,v)^{-r(k+t_i)})^{w_i}\\
&\quad\quad = e(g,w)^{r\lambda_i w_i}\\
&for\ \rho(i) = x':\\
&\quad B_i = (e(C'_{i,1}, K_1)e(C'_{i,2}, K_{i,2})e(C'_{i,3}, K'_{i,3}))^{w_i}\\
&\quad\quad = (e(w^{\lambda_i} v^{t_i} v^k, g^r)e((u^{\rho(i)}h)^{-t_i}(u^{\rho(i)}h)^{-k}, g^{r_i})e((g^{t_i} g^k)^{1/v_{\rho(i)}}, ((u^{x'}h)^{r_i} v^{-r})^{v_{x'}}))^{w_i}\\
&\quad\quad = (e(w,g)^{r\lambda_i} e(v,g)^{r(k+t_i)} e(u^{\rho(i)}h, g)^{r_i(-t_i-k)} e(g, u^{A_i} h)^{r_i(t_i+k)} e(g,v)^{-r(k+t_i)})^{w_i}\\
&\quad\quad = e(g,w)^{r\lambda_i w_i}
\end{aligned}
$$

$$\mathrm{A} = \prod_{i\in I} B_i = \prod_{i\in I} e(g,w)^{r\lambda_i w_i} = e(g,w)^{rs}$$

Then we have

$$
\begin{aligned}
\frac{C' \cdot \mathrm{A}}{e(C'_1, SK'_2)\cdot e(C'_0, K_0)} &= \frac{m\cdot e(g,g)^{\alpha s}\cdot e(g,w)^{rs}}{e(g^{s/k}, (g^{\alpha_2})^k)\cdot e(g^s, g^{\alpha_1} w^r)}\\
&= \frac{m\cdot e(g,g)^{\alpha s}\cdot e(g,w)^{rs}}{e(g,g)^{\alpha_2 s}\cdot e(g,g)^{\alpha_1 s} e(g,w)^{rs}}\\
&= m
\end{aligned}
$$

C. CPA Security

Theorem 1: If the decisional q-type assumption holds in $\mathbb{G}$ and $\mathbb{G}_T$, then there exists no polynomial time attacker to break our revocable large universe CP-ABE scheme selectively, where the challenge matrix is $\mathbf{M}^*(l^* \times n^*)$ with $l^*, n^* \leq q$.

Proof: If there exists an attacker A who can selectively break our proposed CP-ABE scheme with a non-negligible advantage $\varepsilon = Adv_A$, then we can construct a challenger B to break the decisional q-type assumption successfully.

Init: The challenger B takes as input a q-type challenge $\vec{y}, T$. In addition, the attacker A gives the challenge access control policy $(\mathbf{M}^*, \rho^*)$ and the revocation users list RL_{x^*} of attribute x^*. We have that M^* is an $l^* \times n^*$ matrix, where $l^*, n^* \leq q$ and $\rho^* : [l] \to Z_P$.

Setup: The challenger B chooses random exponents $\alpha', \alpha'' \in Z_p$ and implicitly sets $\alpha_1 = \alpha' + a^{q+1}$, $\alpha_2 = \alpha''$, $\alpha = \alpha' + a^{q+1} + \alpha''$ by setting $e(g,g)^\alpha = e(g^a, g^{a^q})\cdot e(g,g)^{\alpha'}\cdot e(g,g)^{\alpha''}$. Note that this way α is correctly distributed and a is information-theoretically hidden from the attacker A. Then B chooses random exponents $u', v', h' \in Z_P$ and using the assumption to construct the following public keys:

$$
\begin{aligned}
&g = g, u = g^{u'} \cdot \prod_{(j,k)\in[l,n]} (g^{a^k/b_j^2})^{\mathbf{M}^*_{j,k}}\\
&h = g^{h'} \cdot \prod_{(j,k)\in[l,n]} (g^{a^k/b_j^2})^{-\rho^*(j)\mathbf{M}^*_{j,k}}\\
&w = g^a, v = g^{v'} \cdot \prod_{(j,k)\in[l,n]} (g^{a^k/b_j})^{\mathbf{M}^*_{j,k}}\\
&e(g,g)^\alpha = e(g^a, g^{a^q})\cdot e(g,g)^{\alpha'}\cdot e(g,g)^{\alpha''}
\end{aligned}
$$

The parameter w is distributed correctly in A's view because the component $e(g,g)^\alpha$ hides the exponent a. Moreover, the terms u, v, h are also distributed correctly because of the randomness of u', v', h'.

Query Chase 1: A makes to B a series of queries including the key generation query $\mathcal{O}_{kg}$ and the re-encryption query $\mathcal{O}_{ree}$.

• $\mathcal{A}$ makes to B a key generation query $\mathcal{O}_{kg}$ associated with the non-authorized attributes set S, since S is non

authorized for $(\mathbf{M}^*, \rho^*)$, there exists a vector $\vec{\mathbf{w}} = (w_1, ..., w_{n^*}) \in Z_p^{n^*}$ where $w_1 = -1$, and for all $\rho^*(i) \in S$, it satisfies $\mathbf{M}_i^* \vec{\mathbf{w}}^T = 0$. Then B chooses a random parameter $t \in Z_p$ and defines the exponent r as:

$$r = t + w_1 a^q + w_2 a^{q-1} + ... + w_{n^*} a^{q+1-n^*} = r + \sum_{i \in [n^*]} w_i a^{q+1-i}.$$

Note that r is distributed correctly because of the randomness of t. Next, B computes the key components as follows:

$$K_1 = g^r = g^{(t + w_1 a^q + w_2 a^{q-1} + ... + w_{n^*} a^{q+1-n^*})} = g^t \prod_{i \in [n^*]} (g^{a^{q+1-i}})^{w_i}$$

According to the definition of t and $w_1 = -1$, we know that w^r includes the item $g^{-a^{q+1}}$. Although $g^{-a^{q+1}}$ is not given in the assumption, it can be canceled by multiplying w^r with $g^{\alpha_1} = g^{\alpha'} g^{a^{q+1}}$, because we implicitly set $\alpha_1 = \alpha' + a^{q+1}$ when generating the key component K_1. In detail, it is constructed as follows:

$$K_1 = g^{\alpha_1} w^r = g^{\alpha'} g^{a^{q+1}} g^{at} \prod_{i \in [n^*]} (g^{a^{q+2-i}})^{w_i} = g^{\alpha'} (g^a)^t \prod_{i=2}^{n^*} (g^{a^{q+2-i}})^{w_i}$$

Next, B will compute the key component $K_{\tau,2}, K_{\tau,3}, \forall \tau \in S$. Before this, it will first sets the common part v^{-r} as follows:

$$\begin{aligned} v^{-r} &= v^{-t} (g^{v'} \prod_{(j,k) \in [l^*, n^*]} g^{a^k \boldsymbol{M}_{j,k}^* / b_j})^{-\sum_{i \in [n^*]} w_i a^{q+1-i}} \\ &= v^{-t} \prod_{i \in [n^*]} (g^{a^{q+1-i}})^{-v' w_i} \cdot \prod_{(i,j,k) \in [n^*, l^*, n^*]} g^{-w_i \boldsymbol{M}_{j,k}^* a^{q+1+k-i} / b_j} \\ &= v^{-t} \prod_{i \in [n^*]} (g^{a^{q+1-i}})^{-v' w_i} \cdot \prod_{(i,j,k) \in [n^*, l^*, n^*], i \neq k} (g^{a^{q+1+k-i} / b_j})^{-w_i \boldsymbol{M}_{j,k}^*} \\ &\quad \cdot \prod_{(i,j) \in [n^*, l^*]} (g^{a^{q+1} / b_j})^{-w_i \boldsymbol{M}_{j,i}^*} \end{aligned}$$

Let

$$v^{-t} \prod_{i \in [n^*]} (g^{a^{q+1-i}})^{-v' w_i} \cdot \prod_{(i,j,k) \in [n^*, l^*, n^*], i \neq k} (g^{a^{q+1+k-i} / b_j})^{-w_i \boldsymbol{M}_{j,k}^*} = \varphi,$$

then we have

$$\begin{aligned} v^{-r} &= \varphi \cdot \prod_{(i,j) \in [n,l]} (g^{a^{q+1} / b_j})^{-w_i \boldsymbol{M}_{j,i}^*} \\ &= \varphi \cdot \prod_{j \in [l]} (g^{-<w, \boldsymbol{M}_j^*>})^{a^{q+1} / b_j} \\ &= \varphi \cdot \prod_{j \in [l], \rho^*(j) \notin S} (g^{-<w, \boldsymbol{M}_j^*>})^{a^{q+1} / b_j} \end{aligned}$$

Note that, B can compute the part φ by using the parameters given in the assumption, while the remaining part has to be canceled by the term $(u^{A_\tau} h)^{r_\tau}$. Therefore, for each attribute $A_\tau \in S$, B chooses a random parameter $r'_\tau \in Z_p$ and implicitly sets

$$\begin{aligned} r_\tau &= r'_\tau + r \cdot \sum_{i' \in [l], \rho^*(i') \notin S} b_{i'} / (A_\tau - \rho^*(i')) \\ &= r'_\tau + t \cdot \sum_{i' \in [l], \rho^*(i') \notin S} b_{i'} / (A_\tau - \rho^*(i')) \\ &\quad + \sum_{(i,i') \in [n,l], \rho^*(i') \notin S} w_i a^{q+1-i} b_{i'} / (A_\tau - \rho^*(i')) \end{aligned}$$

Then B can compute the term $(u^{A_\tau} h)^{r_\tau}$ of key component $K_{\tau,3}$ as

$$\begin{aligned} (u^{A_\tau} h)^{r_\tau} &= (u^{A_\tau} h)^{r'_\tau} \cdot (K_{\tau,2} / g^{r'_\tau})^{u' A_\tau + h'} \\ &\quad \cdot \prod_{(i',j,k) \in [n,l,n], \rho^*(i') \notin S} (g^{t(A_\tau - \rho^*(j)) \boldsymbol{M}_{j,k}^* b_{i'} a^k / (A_\tau - \rho^*(i') b_j^2)}) \\ &\quad \cdot \prod_{(i,i',j,k) \in [n,l,l,n], \rho^*(i') \notin S} (g^{(A_\tau - \rho^*(j)) w_i \boldsymbol{M}_{j,k}^* b_{i'} a^{q+1+k-i} / (A_\tau - \rho^*(i') b_j^2)}) \\ &= \varphi \cdot \prod_{(i,j) \in [n,l], \rho^*(j) \notin S} (g^{(A_\tau - \rho^*(j)) w_i \boldsymbol{M}_{j,i}^* b_j a^{q+1+i-i} / (A_\tau - \rho^*(j) b_j^2)} \\ &= \varphi \cdot \prod_{j \in [l], \rho^*(j) \notin S} g^{<w, \boldsymbol{M}_j^*> a^{q+1} / b_j} \end{aligned}$$

where φ includes the remaining terms of the product. The terms φ and $K_{\tau,2}$ can be computed by using the parameters given in the assumption. The second term of $(u^{A_\tau} h)^{r_\tau}$ cancels exactly with the problematic term of v^{-r}. Therefore, B can compute the key components $K_{\tau,2}$ and $K_{\tau,3}$. Finally, B sets the secret key $SK_1 = (S, K_0, K_1, \{K_{\tau,2}, K_{\tau,3}\}_{\tau \in S})$ and sends SK_1 to the attacker A.

A makes to B a re-encryption query $\mathcal{O}_{ree}$ associated with the revocation users list $RL_{x'}$ of attribute x' and the ciphertext $CT = (A, C, C_0, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [l]})$. Then B generates the re-encrypted ciphertext as follows:

If there is no attribute revoked, namely $RL_{x'} = \Phi$, then the CSP chooses a random $k \in Z_p$ and re-encrypts the ciphertext CT as follows:

$$C' = C = m \cdot e(g,g)^{\alpha s}, C'_0 = C_0 = g^s, C'_1 = (C_0)^{1/k} = g^{s/k},$$
$$\forall i = 1, 2, ..., l:$$

$$\begin{aligned} C'_{i,1} &= C_{i,1} \cdot v^k = w^{\lambda_i} v^{t_i} v^k, C'_{i,2} = C_{i,2} \cdot (u^{\rho(i)} h)^{-k} \\ &= (u^{\rho(i)} h)^{-t_i} (u^{\rho(i)} h)^{-k}, C'_{i,3} = C_{i,3} \cdot g^k = g^{t_i} g^k \end{aligned}$$

Therefore, the re-encrypted ciphertext is set as $CT' = (C', C_0', C_1', \{C_{i,1}', C_{i,2}', C_{i,3}'\}_{i=1}^{l})$.

If there is an attribute x' revoked from a user ID_j, namely $RL_{x'} \neq \Phi$, then the CSP will chooses a random exponent $v_{x'} \in Z_p$ and encrypt it as $\hat{C}$ under the access structure $(\boldsymbol{M}^*, \rho^*)$. Then the CSP also chooses a random $k \in Z_p$ and re-encrypts the ciphertext CT as follows:

$$C' = C = m \cdot e(g,g)^{\alpha s}, C_0' = C_0 = g^s, C_1' = (C_0)^{1/k} = g^{s/k}$$

$$\forall i = 1, 2, ..., l:$$

$$C_{i,1}' = C_{i,1} \cdot v^k = w^{\lambda_i} v^{t_i} v^k, C_{i,2}' = C_{i,2} \cdot (u^{\rho(i)} h)^{-k} = (u^{\rho(i)} h)^{-t_i} (u^{\rho(i)} h)^{-k},$$
$$for\ \rho(i) \neq x: C_{i,3}' = C_{i,3} \cdot g^k = g^{t_i} g^k$$
$$for\ \rho(i) = x: C_{i,3}' = (C_{i,3} \cdot g^k)^{1/v_{\rho(i)}} = (g^{t_i} g^k)^{1/v_{\rho(i)}}$$

Therefore, the re-encrypted ciphertext is set as $CT' = (\hat{C}, C', C_0', C_1', \{C_{i,1}', C_{i,2}', C_{i,3}'\}_{i=1}^{l})$.

Finally, B sets CT' to the attacker A.

Challenge: The attacker A submits to the challenger B two messages m_0 and m_1 with the equal length. Then B selects a random coin $\beta \in \{0,1\}$ and generates the challenge ciphertext components as:

$$C^* = m_\beta \cdot T \cdot e(g^s, g^{\alpha'}) \cdot e(g^s, g^{\alpha''}), C_0^* = g^s$$

Next, B selects random parameters $y_2', ..., y_{n^*}' \in Z_p$, and then sets the vector $\vec{\mathbf{v}} = (s, sa + y_2', sa^2 + y_3', ..., sa^{n-1} + y_{n^*}') \in Z_p^{n^*}$ to implicitly share the key s. Since $\vec{\lambda} = \boldsymbol{M}^* \vec{v}$, we have

$$\lambda_\tau = \sum_{i \in [n]} \boldsymbol{M}_{\tau,i}^* sa^{i-1} + \sum_{i=2}^{n} \boldsymbol{M}_{\tau,i}^* y_i'$$

Let $\lambda_\tau' = \sum_{i=2}^{n} \boldsymbol{M}_{\tau,i}^* y_i'$ and λ_τ' are known to B. For each row, B implicitly sets $t_\tau = -sb_\tau$. Next, it continues to compute:

$$C_{\tau,1} = w^{\lambda_\tau} v^{t_\tau} = w^{\lambda_\tau'} \cdot \prod_{i \in [n]} g^{\boldsymbol{M}_{\tau,i}^* sa^{i-1}} \cdot (g^{sb_\tau})^{-v'} \cdot \prod_{(j,k) \in [l,n]} g^{-\boldsymbol{M}_{j,k}^* a^k sb_\tau / b_j}$$
$$= w^{\lambda_\tau'} \cdot (g^{sb_\tau})^{-v'} \cdot \prod_{(j,k) \in [l,n], j \neq \tau} (g^{a^k sb_\tau / b_j})^{-\boldsymbol{M}_{j,k}^*}$$
$$C_{\tau,2} = (u^{\rho^*(\tau)} h)^{-t_\tau} = (g^{sb_\tau})^{-(u' \rho^*(\tau) + h')} \cdot (\prod_{(j,k) \in [l,n]} g^{(\rho^*(\tau) - \rho^*(j)) \boldsymbol{M}_{j,k}^* a^k / b_j^2})^{-sb_\tau}$$
$$= (g^{sb_\tau})^{-(u' \rho^*(\tau) + h')} \cdot \prod_{(j,k) \in [l,n], j \neq \tau} (g^{sb_\tau a^k / b_j^2})^{-(\rho^*(\tau) - \rho^*(j)) \boldsymbol{M}_{j,k}^*}$$
$$C_{\tau,3} = g^{t_\tau} = (g^{sb_\tau})^{-1}$$

Finally, B sends the challenge ciphertext $CT^* = ((\boldsymbol{M}^*, \rho), C, C_0, \{C_{\tau,1}, C_{\tau,2}, C_{\tau,3}\}_{\tau \in [l]})$ to the attacker A.

Query Chase 2: B continues to make to A a series of queries including the key generation query O_{kg} and the ciphertext re-encryption query $\mathcal{O}_{ree}$ as in **Query Chase 1.**

Guess: The attacker A outputs its guess β' for β. If $\beta = \beta'$, then B outputs 0 denoting $T = e(g,g)^{\alpha^{q+1} s}$, otherwise outputs 1 denoting T is a random parameter in $\mathbb{G}_T$. Therefore, if A breaks the game with a non-negligible advantage, then B has a non-negligible advantage to break the q-type assumption.

IV. ANALYSIS

A. Security Requirements

1) Data confidentiality:

In our scheme, if the attributes set of a user does not satisfy the access policy associated with the ciphertext, then the user cannot recover $e(g,w)^{rs}$ during the initial decryption process, thus the ciphertext cannot be decrypted successfully. In addition, when a user is revoked from some attribute groups, then he cannot decrypt the ciphertext $\hat{C}$ to obtain the key material for updating the corresponding key. Therefore, the revoked user cannot decrypt the ciphertext. Finally, we assume the CSP is honest but curious. However, it is only authorized to re-encrypt the ciphertext and cannot obtain the decryption key. Thus, the CSP cannot decrypt the ciphertext also.

2) Collusion resistance:

Assume an attacker does not hold the attribute $\rho(i)$. According to the previous CP-ABE schemes, we know that the secret key s is embedded in the ciphertext rather than the key. In order to decrypt the ciphertext, a colluding attacker needs to obtain $e(g,w)^{rs}$. However, the attacker must pair $C_{i,1}, C_{i,2}, C_{i,3}$ from the ciphertext and $K_{i,2}$, $K_{i,3}$ from other

colluding attacker's secret key for the attribute, but $e(g,w)^{rs}$ cannot be obtained by doing so since user's key is uniquely generated by a random exponent r.

3) Forward security:

Assume a user is revoked from an attribute group, namely he cannot decrypt the ciphertext $\hat{C}$ to obtain the key material for updating the corresponding key. Therefore, the revoked user cannot decrypt any components corresponding the attributes after his revocation.

4) Backward security:

Assume a user joins an attribute group that satisfy the access policy associated with the ciphertext. Even if the user can update his secret key and obtains the previous ciphertext, he still cannot decrypt the ciphertext successfully. That is because the user's secret key is updated with a new random exponent $v_{\rho(i)}$, but the component of the ciphertext $C'_{i,3} = (g^{t_i} g^k)^{1/v_{\rho(i)}}$ is re-encrypted with the previous random $v_{\rho(i)}$ and k.Therefore, the backward security of the shared data is achieved in our proposed scheme.

B. Functionality

TABLE I. COMPARISON OF FUNCTIONALITY

Scheme	universe	Model	Assumption
Liang	small	standard	DBDH
Hur	small	generic group	-
Yang	small	random oracle	q-parallel BDHE q-type
Ours	large	standard	

The functionality comparison is demonstrated in Table 1, from which can see that Liang's scheme, Hur's scheme and Yang's scheme achieve small universe, namely the size of the attribute space is bounded with the security parameter and attributes are fixed at setup, moreover, the size of the public parameters grows linearly with the number of attributes. However, our scheme achieves large universe, namely the size of the attribute universe can be exponentially large, which is a desirable feature. In addition, compared with the generic group model of Hur's scheme and the random oracle model of Yang's scheme, Liang's scheme and our scheme are provably secure based on the stand assumption in the standard model, which have stronger security.

REFERENCES

[1] Sahai A, Waters B. Fuzzy Identity-Based Encryption[C]// International Conference on Theory and Applications of Cryptographic Techniques. Springer-Verlag, 2005:457-473.

[2] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data.[J]. Proc of Acmccs', 2010, 89-98:89-98.

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-Policy Attribute-Based Encryption[J]. 2007, 2008(4):321-334.

[4] Goyal V, Jain A, Pandey O, et al. Bounded Ciphertext Policy Attribute Based Encryption[M]//Automata, Languages and Programming. 2015:579-591.

[5] Lewko A, Okamoto T, Sahai A, et al. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption[C]// Advances in Cryptology - EUROCRYPT 2010, International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings. 2010:62-91.

[6] Attrapadung N, Libert B, Panafieu E D. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts[C]// Public Key Cryptography - PKC 2011 -, International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings. 2011:90-108.

[7] Wang C, Luo J. An Efficient Key-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length[J]. Mathematical Problems in Engineering, 2013, 2013(3):87-118.

[8] Lai J, Deng R H, Li Y, et al. Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption[C]// ACM Symposium on Information, Computer and Communications Security. ACM, 2014:239-248.

[9] Lewko A, Waters B. Unbounded HIBE and Attribute-Based Encryption[C]// International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology. Springer-Verlag, 2011:547-567.

[10] Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption[C]// ACM Sigsac Conference on Computer & Communications Security. ACM, 2013:463-474.

[11] Yu S, Wang C, Ren K, et al. Attribute based data sharing with attribute revocation[C]// ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April. 2010:261-270.

[12] Xie X, Ma H, Li J, et al. New Ciphertext-Policy Attribute-Based Access Control with Efficient Revocation[C]// International Conference on Information and Communication Technology. 2013:373-382.

[13] Yang K, Jia X. Security for Cloud Storage Systems[M].Springer Publishing Company, Incorporated, 2013.

[14] Beimel A. Secure Schemes for Secret Sharing and Key Distribution[J]. International Journal of Pure & Applied Mathematics, 1996.