J. Shanghai Jiaotong Univ. (Sci.), 2011, 16(4): 490-496
DOI: 10.1007/s12204-011-1178-4

# Fully Secure Revocable Attribute-Based Encryption

*QIAN Jun-lei* (钱俊磊), *DONG Xiao-lei** (董晓蕾)
(Department of Computer Science & Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

**Abstract:** Distributed information systems require complex access control which depends upon attributes of protected data and access policies. Traditionally, to enforce the access control, a file server is used to store all data and act as a reference to check the user. Apparently, the drawback of this system is that the security is based on the file server and the data are stored in plaintext. Attribute-based encryption (ABE) is introduced first by Sahai and Waters and can enable an access control mechanism over encrypted data by specifying the users' attributes. According to this mechanism, even though the file server is compromised, we can still keep the security of the data. Besides the access control, user may be deprived of the ability in some situation, for example paying TV. More previous ABE constructions are proven secure in the selective model of security that attacker must announce the target he intends to attack before seeing the public parameters. And few of previous ABE constructions realize revocation of the users' key. This paper presents an ABE scheme that supports revocation and has full security in adaptive model. We adapt the dual system encryption technique recently introduced by Waters to ABE to realize full security.

**Key words:** attribute-based encryption (ABE), dual encryption, revocation
**CLC number:** TP 309.7 **Document code:** A

## 0 Introduction

In order to gain a much larger world of possibilities for sharing encrypted data, attribute-based encryption (ABE) is introduced in fuzzy identity-based encryption (IBE)[1]. ABE enables an access control mechanism over encrypted data by using access policies. Subsequently, Goyal et al.[2] formulated two complimentary forms of ABE: ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). In the CP-ABE system, keys are associated with sets of attributes and ciphertexts are associated with access policies[3-5]. In the KP-ABE system, encrypted data are associated with a set of attributes. An authority will issue users different private keys which are associated with a different access structure over attributes and reflect the access policies. The decryption algorithm allows users to decrypt data if the access policy of their private key permits the attributes. KP-ABE is thought to be useful in some broadcast situation. For example in online video, publisher can distribute some video with attribute {"24 hours", "session", "VIP user"} and user that has the access policy such as {"general user" OR ("24 hours" AND "Prison Break")} can decrypt the ciphertext and then get the film data.

The previous proof of ABE is usually selectively secure that attacker is required to announce the target he intends to attack before seeing the public parameters. This is a partitioning strategy that the previous proof builds the hard problem like "decisional bilinear Diffie-Hellem" (DBDH)[6] problem into the public parameters and target keys that cannot be queried by attacker. The simulator can only make queries of the rest of the key space.

Encryption schemes need revocation mechanism when they face the key compromise and key expiration problem. In public key infrastructure (PKI), revocation is generally implemented via certificate revocation lists (CRLs). In attribute-based setting, Boldyreva et al.[7] proposed a revocable ABE scheme by update the non-revoked users at all time slots. Huge communication from the key authority to the users is its drawback.

## 1 The Proposed Approach

We adapt the dual system encryption technique to ABE in order to gain full security. Waters introduced dual system encryption to overcome the limitations of partitioning[8]. In a dual encryption system, ciphertexts and keys have two forms: normal and semi-functional. A normal key can decrypt normal or semi-functional ciphertexts. A semi-functional key can only

**Received date:** 2010-11-04
**Foundation item:** the National Natural Science Foundation of China (No. 60972034)
***E-mail:** dong-xl@cs.sjtu.edu.cn

decrypt normal ciphertexts. When a semi-functional key is used to decrypt a semi-functional ciphertext, decryption fails with a random additional term from pairing of the terms in the semi-functional key and ciphertext. The semi-functional ciphertexts and keys are only used in the proof of security other than in the real system. The proof employs a sequence of security games that the ciphertext and queried keys are changed to semi-functional one by one. In the final game, none of the key given to the attacker can be useful for decrypting a semi-functional ciphertext. The challenge is that in the step where the $k$th key becomes semi-functional, the simulator must be prepared to make any semi-functional challenge ciphertext and any key as the $k$th key. This means that the simulator can just make a key that should decrypt the challenge ciphertext and test the key itself that whether the key is semi-functional by attempting to decrypt the semi-functional challenge ciphertext. Lewko and Waters[9-10] provided a realization of dual system to overcome it by using nominally semi-functional keys. Nominally semi-functional keys are almost the same as the semi-functional keys except that they also successfully decrypt the semi-functional ciphertexts. We adapt this technique in our construction to realize full security. The proof of the system relies on the complexity assumptions of composite groups introduced in Ref. [9]. Revocation list is needed in encryption, so it has application similar to broadcast that user may not have the access control policy, but just a set of attributes. For that reason, in our construction we use KP-ABE structure. Additional, we add the binary tree[11-12] technique to realize the key revocation.

## 2 Preliminaries

**Access structures** Let $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$ be a set of parties, $\mathcal{P}_i$ means the $i$th party. A collection $\mathbb{A} \subseteq 2^{\{\mathcal{P}_1,\mathcal{P}_2,\cdots,\mathcal{P}_n\}}$ is monotone for $\forall B, C$ : if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection $\mathbb{A}$ of non-empty subsets of $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$, i.e. $\mathbb{A} \subseteq 2^{\{\mathcal{P}_1,\mathcal{P}_2,\cdots,\mathcal{P}_n\}}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

**Linear secret-sharing schemes (LSSS)** A secret-sharing scheme II is called linear (over $\mathbf{Z}_p$) over a set of parties $\mathcal{P}$ if ① the shares for each party form a vector over $\mathbf{Z}_p$; ② there exists a matrix $\boldsymbol{M}$ called the share-generating matrix for the secret-sharing scheme II. The matrix $\boldsymbol{M}$ has $l$ rows and $n$ columns. For all $i = 1, 2, \cdots, l$, we denote the function $\rho(i)$ as the party labeling row $i$ of $\boldsymbol{M}$. When we consider the column vector $\boldsymbol{v} = [s\ \ r_2\ \ r_3\ \ \cdots\ \ r_n]^{\mathrm{T}}$, where $s \in \mathbf{Z}_p$ is the secret to be shared, and $r_2, r_3, \cdots, r_n \in \mathbf{Z}_p$ are randomly chosen, then $\boldsymbol{Mv}$ is the share $l$ vector of the secret $s$ according to the secret-sharing scheme II. The share $(Mv)_i$ belongs to party function $\rho(i)$. In the later scheme we define the symbol $\mathrm{AS}(\boldsymbol{M}, \rho)$ for a secret-sharing scheme II that has a share-generating matrix $\boldsymbol{M}$ and function $\rho$.

LSSS enjoys the linear reconstruction property. Suppose that the secret-sharing scheme II is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subseteq \{1, 2, \cdots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. If $\{\lambda_i\}$ are valid shares of any secret $s$ according to secret-sharing scheme II, then there exist constants $\{\omega_i \in Z_p\}_{i \in I}$ satisfy $\sum\limits_{i \in I} \omega_i \lambda_i = s$. Furthermore, it is shown that these constants $\omega_i$ can be found in time polynomial in the size of the share-generating matrix $\boldsymbol{M}$.

**Composite order bilinear groups** We construct our systems in the composite order bilinear groups. We define a group generator $\mathcal{G}$, and an algorithm which takes a security parameter $\lambda$ as input and outputs a description of a bilinear group $G$. For our purposes, we have that $\mathcal{G}$ outputs $(p_1, p_2, p_3, G, G_T, e)$ where $p_1, p_2, p_3$ are distinct primes, $G$ and $G_T$ are cyclic groups of order $N = p_1 p_2 p_3$, and $e : G_2 \to G_T$ is a map.

(1) Bilinear: $\forall g, h \in G; a, b \in \mathbf{Z}_N; e(g^a, h^b) = e(g, h)^{ab}$.

(2) Non-degenerate: $\exists g \in G$ such that $e(g, g)$ has order $N$ in $G_T$.

We assume that the group operations in $G$ and $G_T$ as well as the bilinear map $e$ are computable in polynomial time with respect to $\lambda$ and that the group description of $G$ and $G_T$ includes generators of the respective cyclic groups. We let $G_{p_1}$, $G_{p_2}$ and $G_{p_3}$ denote subgroups of order $p_1$, $p_2$ and $p_3$ in $G$ respectively. We note that when $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, $e(h_i, h_j)$ is the identity element in $G_T$. To prove this, we suppose $h_1 \in G_{p_1}$ and $h_2 \in G_{p_2}$. Let $g$ denote a generator of $G$. Then, $g^{p_1p_2}$ generates $G_{p_3}$, $g^{p_1p_3}$ generates $G_{p_2}$, and $g^{p_2p_3}$ generates $G_{p_1}$. Hence, for some $\alpha_1$ and $\alpha_2$ which satisfy $h_1 = (g^{p_2p_3})^{\alpha_1}$ and $h_2 = (g^{p_1p_3})^{\alpha_2}$, there is

$$\begin{aligned} e(h_1, h_2) = e(g^{p_2p_3\alpha_1}, g^{p_1p_3\alpha_2}) = \\ e(g^{\alpha_1}, g^{p_3\alpha_2})^{p_1p_2p_3} = 1. \end{aligned}$$

This orthogonality property of $G_{p_1}$, $G_{p_2}$ and $G_{p_3}$ is used to implement semi-functionality in our constructions.

The proof of the security relies on the complexity assumptions of composite groups. The subgroup decision problem for 3 primes is below. We define the symbol $g \xleftarrow{\mathrm{R}} G_{p_1}$ for the variable $g$ is selected randomly ($\xleftarrow{\mathrm{R}}$) from the group $G_{p_1}$.

**Assumption 1** Given a group generator $\mathcal{G}$, we define the following distributions

$$G = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G},$$
$$g \xleftarrow{\mathrm{R}} G_{p_1}, \quad X_3 \xleftarrow{\mathrm{R}} G_{p_3},$$
$$D = (G, g, X_3), \quad T_1 \xleftarrow{\mathrm{R}} G_{p_1p_2}, \quad T_2 \xleftarrow{\mathrm{R}} G_{p_1}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be

$$\mathrm{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda) := |P[\mathcal{A}(D, T_1) = 1] - P[\mathcal{A}(D, T_2) = 1]|.$$

where $P$ is the possibility function.

**Assumption 2** Given a group generator $\mathcal{G}$, we define the following distributions

$$G = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G},$$
$$g \xleftarrow{\mathrm{R}} G_{p_1}, \quad (X_2, Y_2) \xleftarrow{\mathrm{R}} G_{p_2}, \quad (X_3, Y_3) \xleftarrow{} G_{p_3}S,$$
$$D = (G, g, X_1X_2, X_3, Y_2Y_3),$$
$$T_1 \xleftarrow{\mathrm{R}} G, \quad T_2 \xleftarrow{\mathrm{R}} G_{p_1p_2}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be

$$\mathrm{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) := |P[\mathcal{A}(D, T_1) = 1] - P[\mathcal{A}(D, T_2) = 1]|.$$

**Assumption 3** Given a group generator $\mathcal{G}$, we define the following distributions

$$G = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{\mathrm{R}} \mathcal{G}, (\alpha, s) \xleftarrow{\mathrm{R}} \mathbf{Z}_N,$$
$$g \xleftarrow{\mathrm{R}} G_{p_1}, \quad (X_2, Y_2, Z_2) \xleftarrow{\mathrm{R}} G_{p_2}, \quad X_3 \xleftarrow{\mathrm{R}} G_{p_3},$$
$$D = (G, g, g^{\alpha}X_2, X_3, g^sY_2, Z_2),$$
$$T_1 = e(g, g)^{\alpha s}, \quad T_2 \xleftarrow{\mathrm{R}} G_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be

$$\mathrm{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) := |P[\mathcal{A}(D, T_1) = 1] - P[\mathcal{A}(D, T_2) = 1]|.$$

**Terminologies for binary tree** We denote some terminology for complete binary tree. Let $\mathcal{L} = \{1, 2, \cdots, n\}$ be the set of leaves. Let $\mathcal{X}$ be the set of node names in the tree via some systematic naming order. For a leaf $i \in \mathcal{L}$, let $\mathrm{Path}(i) \subseteq \mathcal{X}$ be the set of all nodes on the path from node $i$ to the root (including $i$ and the root). For leaf node set $L \subseteq \mathcal{L}$, let $\mathrm{Cover}(L) \subseteq \mathcal{X}$ be defined as follows. First mark all the nodes in $\mathrm{Path}(i)$ if $i \in L$. Then $\mathrm{Cover}(L)$ is the set of all the unmarked children of marked nodes. It can be shown to be the minimal set that contains no node in $\mathrm{Path}(i)$ if $i \in L$ but contains at least one node in $\mathrm{Path}(i)$ if $i \notin L$. It is known in Ref. [13] that

$$|\mathrm{Cover}(L)| \leqslant |L|[\log(n/|L|) + 1].$$

## 3 Definition

**Setup** $(\lambda) \rightarrow (\mathrm{PK}, \mathrm{MSK})$ In our scheme, we define the universe set of serial numbers $\mathcal{U}$ as the set of leaves in the complete binary tree $\mathcal{L} = \{1, 2, \cdots, n\}$. Setup is a randomized algorithm that takes an input $\lambda$ with a size of $\mathcal{U}$. It outputs the public key PK and master key MSK.

**Encrypt** $(\mathrm{PK}, M, R, \omega) \rightarrow \mathrm{CT}$ This is a randomized algorithm that takes in public parameters PK, message $M$, set of attributes $\omega$, and revocation list $R$ (a list of keys that have been revocated). It outputs ciphertext CT.

**Key-Gen** $(\mathrm{ID}, \mathrm{AS}(\boldsymbol{A}, \rho), \mathrm{MSK}, \mathrm{PK}) \rightarrow \mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)}$ This is a randomized algorithm that takes serial number $\mathrm{ID} \in \mathcal{U}$, access structure $\mathrm{AS}(\boldsymbol{A}, \rho)$ over the universe of attribute $i$ of matrix $\boldsymbol{A}$ to an attribute $\rho(i)$, master key MSK, and public key PK. It outputs private decryption key $\mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)}$.

**Decrypt** $(\mathrm{CT}, \omega, R, \mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)}, (\mathrm{ID},\mathrm{AS}(\boldsymbol{A}, \rho)),\mathrm{PK})$ This algorithm takes as input the ciphertext CT that is encrypted under attributes $\omega$, revocation list $R$, decryption key $\mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)}$ for user serial number ID with access control structure $\mathrm{AS}(\boldsymbol{A}, \rho)$, and public key PK. It outputs the message $M$ or a special symbol $\perp$ indicating an unsuccessful decryption.

**Security model** We now give the security model for revocable KP-ABE systems. This is described by a security game between a challenger and an attacker. The game proceeds as follows.

**Setup** The challenger runs the Setup algorithm and gives the public parameters PK to the attacker.

**Phase 1** The attack queries the challenger for private keys corresponding to an access structure

$$\mathrm{AS}(\boldsymbol{A}, \rho)_1, \mathrm{AS}(\boldsymbol{A}, \rho)_2, \cdots, \mathrm{AS}(\boldsymbol{A}, \rho)_{q_1}$$

**Challenge** The attacker declares two equal length messages $M_0$ and $M_1$, an attributes set $\omega$, and a revocation list $R$. The access structures $\mathrm{AS}(\boldsymbol{A}, \rho)_1$, $\mathrm{AS}(\boldsymbol{A}, \rho)_2$, $\cdots$, $\mathrm{AS}(\boldsymbol{A}, \rho)_{q_1}$ which are not included in the $R$ cannot be satisfied by the attributes' set $\omega$. The challenger flips random coin $\beta \in \{0, 1\}$, and encrypts message $M_\beta$, producing $\mathrm{CT}^*$ to the simulator CT. It gives $\mathrm{CT}^*$ to the attacker.

**Phase 2** The attack queries the challenger for private keys corresponding to an access structure $\mathrm{AS}(\boldsymbol{A}, \rho)_{q_1+1}, \mathrm{AS}(\boldsymbol{A}, \rho)_{q_1+2}, \cdots, \mathrm{AS}(\boldsymbol{A}, \rho)_q$ with added restriction that if the $i$th key is not in the revocation list $R$, then $\omega$ does not satisfy $\mathrm{AS}(\boldsymbol{A}, \rho)_i$.

**Guess** The attack outputs a guess $\beta'$ for $\beta$. The advantage of an attack in this game is defined to be $P[\beta = \beta'] - \dfrac{1}{2}$.

Selective security is defined by adding an initialization phase where attack must declare $\omega$ in the encryption before seeing PK. In this work, we do not impose this restriction on the attacker. We use the dual system encryption technique to prove the security in this model in the Section 5.

## 4 Construction

Let $\mathcal{U}$ be the universes of user key serial numbers. Let $G_{p_2}$ be the maximum size of attribute set allowed to be associated with a ciphertext, i.e., we restrict $|\omega| \leqslant m$. We set $R$ as the user key set $R \subseteq \mathcal{U}$ and $d$ be the maximum of Cover($R$) for all user key sets $R$. The restriction of $|\omega|$ and $d$ is important in the security proof.

**Setup** $(\lambda) \rightarrow \mathrm{PK}, \mathrm{MSK}$ The setup algorithm chooses a bilinear group $G$ of order $N = p_1p_2p_3$. Let $G_{p_i}$ denote the subgroup of order $p_i$ in $G$. Then it chooses randomly $\alpha \in \mathbf{Z}_N, g \in G_{p_1}$. It chooses randomly $s_i \in \mathbf{Z}_N$ for each attribute $i$. Then it chooses random numbers for $d$ times: $h_0, h_1, \cdots, h_d \in \mathbf{Z}_N$. We set symbols $T_i$, $H_j$ for $T_i = g^{s_i}(\forall i), H_j = g^{h_j}(\forall j)$. Define a function $O(x) = \prod_{j=0}^{d} H_j^{x^j}$. We define MSK= $\{\alpha$, a generator $X_3 \in G_{p_3}\}$ and public keys PK=$\{N, g, e(g,g)^\alpha$, $T_i$, $H_j$, $O(x)\}$.

**Encrypt** $(\mathrm{PK}, M, R, \omega)$ The algorithm first picks randomly $s \in \mathbf{Z}_N$, and computes $C = M(e(g,g)^\alpha)^s$, $C^{(1)} = g^s$, $C_i^{(2)} = T_i^s$, $i \in \omega$. Then we run Cover($R$) to find a minimal node set that covers $\mathcal{U} \backslash R$, and compute for each leaf node

$$x \in \mathrm{Cover}(R) : C_x^{(3)} = O(x)^s.$$

It outputs the ciphertext as

$$\mathrm{CT} = \Big(C, C^{(1)}, \{C_i^{(2)}\}_{i\in\omega}, \{C_x^{(3)}\}_{x\in\mathrm{Cover}(R)}\Big).$$

**Key-Gen** $(\mathrm{ID}, \mathrm{AS}(\boldsymbol{A}, \rho), \mathrm{MSK}, \mathrm{PK}) \rightarrow \mathrm{sk}$ First it chooses random numbers $\alpha_1$ and $\alpha_2$ such that $\alpha = \alpha_1 + \alpha_2$. The key-generation chooses random vector $\boldsymbol{v} = (\alpha_1, v_2, v_3, \cdots, v_k)$ where $v_2, v_3, \cdots, v_k \in \mathbf{Z}_N$. For all leaf nodes $x \in \mathrm{Path}(\mathrm{ID})$, it then randomly chooses $r_1, r_2, \cdots, r_l, r_x \in \mathbf{Z}_N$, and chooses random elements $W_i, V_i, R_x, R_{x'} \in G_{p_3}$. It outputs the private key as

$$\mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)} = \Big(\Big(D_i^{(1)}, D_i^{(2)}\Big)_{i\in[1,l]}, \Big(D_x^{(3)}, D_x^{(4)}\Big)_{x\in\mathrm{Path(ID)}}\Big),$$

where

$$D_i^{(1)} = g^{\boldsymbol{A}_i\cdot\boldsymbol{v}}T_i^{r_i}W_i, \quad D_i^{(2)} = g^{r_i}V_i,$$
$$D_x^{(3)} = g^{\alpha_2}O^{r_x}(x)R_x, \quad D_x^{(4)} = g^{r_x}R_{x'}.$$

**Decrypt** $(\mathrm{CT}, \omega, R, \mathrm{sk}_{\mathrm{ID},\mathrm{AS}(\boldsymbol{A},\rho)}, (\mathrm{ID}, \mathrm{AS}(\boldsymbol{A},\rho)), \mathrm{PK})$ Suppose that $\omega$ satisfies $\mathrm{AS}(\boldsymbol{A},\rho)$ and $\mathrm{ID} \notin R$. Then the algorithm computes constants $c_i$ such that $\sum_{\rho(i)\in\omega} c_i\boldsymbol{A}_i = \boldsymbol{l}$. We define $\boldsymbol{l}$ as vector $(1, 0, \cdots, 0)$. And since $\mathrm{ID} \notin R$, it also finds a node $x$ such that $x \in \mathrm{Path(ID)} \cap \mathrm{Cover}(R)$. Then we compute

$$\prod_{\rho(i)\in\omega}\left(\frac{e(D_i^{(1)}, C^{(1)})}{e(C_{\rho(i)}^{(2)}, D_i^{(2)})}\right)^{c_i}\left(\frac{e(D_x^{(3)}, C^{(1)})}{e(C_x^{(3)}, D_x^{(4)})}\right) =$$
$$e(g,g)^{\alpha_1 s}e(g,g)^{\alpha_2 s} = e(g,g)^{\alpha s},$$
$$M = C/e(g,g)^{\alpha s}.$$

## 5 Security

First we claim that the revocation list $R$ in the ciphertext cannot be corrupted with the definition of public parameter $d$ and $O(x)$. If the user $Y$ in the revocation list can decrypt ciphertext, we must enlarge the Cover($R$) embedded in the $C_x^{(3)}$. With the definition $O(x) = \prod_{j=0}^{d} H_j^{x^j}$, we say that in each ciphertext, $C_x^{(3)}$ is the value in $x$ of a $d$ degree random polynomial with random secret $s$. So with less than $d$ elements $C_{x_j}^{(3)}$ in this polynomial, we cannot construct another value of the polynomial. Then it is impossible to enlarge the Cover($R$), and the revocation list $R$ in the ciphertext is secure. In order to prove the security of our system, we must first define the semi-status of the ciphertexts and the keys.

**Semi-functional ciphertext** We set $g_2$ to be a generator of $G_{p_2}$ and random number $c, \xi_i, \theta_x \in \mathbf{Z}_N$. $C^{(1)} = g^s g_2^c$, $C_i^{(2)} = T_i^s g_2^{c\xi_i}$, $C_x = O^s(x)g^{c\theta_x}$.

**Semi-functional key** We first set $g_2$ to be a generator of $G_{p_2}$ and random number $c, \alpha_2', \xi_i, \varphi_i, \zeta_x, \theta_x, x' \in \mathbf{Z}_N$, where $i \in [1, -1], x \in \mathrm{Path(ID)}$. Then it chooses random vector $\boldsymbol{\lambda}$, and sets $\lambda_i = \boldsymbol{A}_i\boldsymbol{\lambda}$.

Type 1 semi-functional key:

$$D_i^{(1)} = g^{\boldsymbol{A}_i\cdot\boldsymbol{v}}T_i^{r_i}W_i g_2^{\lambda_i+\xi_i\varphi_i}, \quad D_i^{(2)} = g^{r_i}V_i g_2^{\varphi_i},$$
$$D_x^{(3)} = g^{\alpha_2}O^{r_x}(x)R_x g_2^{\alpha_2'+\theta_x\zeta_x}, \quad D_x^{(4)} = g^{r_x}R_{x'}g_2^{\zeta_x}.$$

Type 2 semi-functional key:

$$D_i^{(1)} = g^{\boldsymbol{A}_i\cdot\boldsymbol{v}}T_i^{r_i}W_i g_2^{\lambda_i}, \quad D_i^{(2)} = g^{r_i}V_i,$$
$$D_x^{(3)} = g^{\alpha_2}O^{r_x}(x)R_x g_2^{\alpha_2'}, \quad D_x^{(4)} = g^{r_x}R_{x'}.$$

When a semi-functional key decrypts a semi-ciphertext, there will be additional term

$$e(g_2, g_2)^{\sum\limits_{\rho(i)\in\omega} c\boldsymbol{A}_i c_i\lambda_i} e(g_2, g_2)^{c\alpha_2'}.$$

We say that when $c\boldsymbol{\lambda}\cdot\boldsymbol{l} = 0$ and $\alpha_2' = 0$, we call the type 1 semi-functional key is a nominally semi-functional key because it can still decrypt the semi-ciphertext.

Game$_{\mathrm{real}}$. It is the same as the real security game.

Game$_0$. It is just like the real game except that the ciphertext is changed into semi-functional.

Game$_{k,1}$. The ciphertext is semi-functional. The first $k-1$ queried keys are type 2 semi-functional keys, the $k$th queried key are type 1 semi-functional key, and the left are normal keys.

Game$_{k,2}$. The ciphertext is semi-functional. The first $k$ queried keys are type 2 semi-functional keys, and the left are normal keys.

Game$_{\text{Final}}$. All keys are semi-functional of type 2 and the ciphertext is a semi-functional encryption of random message, independent of the two messages provided by the attacker. These games are indistinguishable under the composite assumption.

**Lemma 1** Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that Game$_{\text{real}}$Adv$_{\mathcal{A}}$ − Game$_0$Adv$_{\mathcal{A}}$ = $\epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage $\epsilon$ in breaking Assumption 1.

**Proof** $\mathcal{B}$ is given as $\{g, X_3, T\}$. It will simulate either Game$_{\text{real}}$ or Game$_0$ with $\mathcal{A}$. The parameters are formed by choosing exponents $\alpha, s_i(\forall i), h_j(\forall j)$ randomly as in the normal construction. $\mathcal{B}$ gives these to $\mathcal{A}$. $\mathcal{B}$ can respond to whatever key requests by running the usual key generation algorithm to make normal keys because it knows all the secret of the system.

To form the challenge ciphertext for a set of attributes $\omega$ and a revocation list $R$, $\mathcal{B}$ implicitly sets $s$ so that $g^s$ is the part of $T$ in the $G_{p_1}$ subgroup (i.e. $T$ is the product of $g^s$ and possibly an element of $G_{p_2}$):

$$C = M_\beta e(g,g)^{s\alpha} = Me(g,T)^\alpha, \quad C^{(1)} = T,$$

$$C_i^{(2)} = T^{s_i}(\forall i \in \omega), \quad C_x^{(3)} = T^{\sum\limits_{j=0}^{d} h_j x^j} \quad (\forall x \in \text{Cover}(R)),$$

where $\beta$ is the random coin, $\beta \in \{0, 1\}$.

We note that this implicitly sets $\xi_i = s_i$ and $\theta_x = \sum\limits_{j=0}^{d} h_j x^j$. First, we say that values of $s_i$ and $h_j$ modulo $p_1$ are uncorrelated from the values of $s_i$ and $h_j$ modulo $p_2$ (for Chinese Remainder Theorem). Furthermore, for $|\text{Cover}(R)| < d$, the attacker cannot use $C_x^{(3)}$ to form another $x$ in the tree. If $T = g^s$, this is a properly distributed normal ciphertext. If $T = g^s X_2$ (for $X_2 \in G_{p_2}$), this is a properly distributed semi-functional ciphertext. Thus, $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage $\epsilon$ in breaking Assumption 1.

**Lemma 2** Suppose that there exists a polynomial time algorithm $\mathcal{A}$ such that Game$_{k-1,2}$Adv$_{\mathcal{A}}$ − Game$_{k,1}$Adv$_{\mathcal{A}}$ = $\epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage $\epsilon$ in breaking Assumption 2.

**Proof** $\mathcal{B}$ is given as $\{g, X_3, g^s X_2, Y_2 Y_3, T\}$. It will simulate either Game$_{k-1,2}$ or Game$_{k,1}$ with $\mathcal{A}$. It begins by sending $\mathcal{A}$ the public parameters $N, g, X_3, e(g,g)^\alpha, g^{s_i}, g_{h_j}$ where it chooses $\alpha$ and values $s_i, h_j$ randomly. In order to form the challenge ciphertext for a set of attribute $\omega$ and revocation list $R$, $\mathcal{B}$ sets

$$C = M_\beta e(g, g^s X_2)^\alpha,$$

$$C^{(1)} = g^s X_2, \quad C_i^{(2)} = (g^s X_2)^{s_i} \quad (\forall i \in \omega),$$

$$C_x^{(3)} = (g^s X_2)^{\sum\limits_{j=0}^{d} h_j x^j} \quad (\forall x \in \text{Cover}(R)).$$

We note that this implicitly sets $\xi_i = s_i$ and $\theta_x = \sum\limits_{j=0}^{d} h_j x^j \pmod N$. We say that values of $s_i$ and $h_j$ modulo $p_1$ are uncorrelated from the values of $s_i$ and $h_j$ modulo $p_2$. In order to form normal keys for queries greater than $k$, $\mathcal{B}$ can use MSK and do the regular key generation algorithm. To create semi-functional keys of type 2 for queries less than $k$, $\mathcal{B}$ chooses random numbers $\alpha_1$ and $\alpha_2$ such that $\alpha = \alpha_1 + \alpha_2 \pmod N$. Then it chooses random vector $\boldsymbol{v}$ such that $\boldsymbol{v} \cdot \boldsymbol{l} = \alpha_1$, random vector $\boldsymbol{v}_2'$, random number $r_i \in \mathbf{Z}_N$, random group elements $W_i, V_i, R_x$ and $R_x'$, and random number $\alpha' \in \mathbf{Z}_N$. The semi-functional key can be defined as

$$D_i^{(1)} = g^{\boldsymbol{A}_i \cdot \boldsymbol{v}} T_i^{r_i} W_i (Y_2 Y_3)^{\boldsymbol{A}_i \cdot \boldsymbol{v}_2'}, \quad D_i^{(2)} = g^{r_i} V_i,$$
$$D_x^{(3)} = g^{\alpha_2} O^{r_x}(x) R_x (Y_2 Y_3)^{\alpha'}, \quad D_x^{(4)} = g^{r_x} R_{x'}.$$

We note that for $Y_2 = g_2^c$, $\boldsymbol{\lambda}$ in our description of semi-functional keys above now corresponds to $\boldsymbol{\lambda} = c\boldsymbol{v}_2'$, and $\alpha_2'$ corresponds to $\alpha_2' = c\alpha'$.

For the $k$th key to AS$(\boldsymbol{A}, \rho)$, $\mathcal{B}$ will make a key that is either a nominally semi-functional key of type 1 or a normal key depending on the value of $T$ in the challenge. We emphasize that a nominally semi-functional key will still have the distribution of a regular semi-functional key of type 1 in the view of an attacker. To form the $k$th key for AS$(\boldsymbol{A}, \rho)$, $\mathcal{B}$ first chooses random numbers $\alpha_1$ and $\alpha_2$ such that $\alpha = \alpha_1 + \alpha_2$ modulo $N$. Then it chooses random vector $\boldsymbol{v}''$ such that $\boldsymbol{v}'' \cdot \boldsymbol{l} = 0$ and a vector $\boldsymbol{v}'$ such that $\boldsymbol{v}' \cdot \boldsymbol{l} = 0$. It implicitly sets $\boldsymbol{v} = r\boldsymbol{v}_2 + \boldsymbol{v}'$, where $g^r$ is the $G_{p_1}$ part of $T$. Elements $\varphi_i, \zeta_x, W_i$ and $V_i$ are chosen randomly.

$$D_i^{(1)} = g^{\boldsymbol{A}_i \cdot \boldsymbol{v}'} T^{\boldsymbol{A}_i \cdot \boldsymbol{v}''} T^{\varphi_i s_{\rho(i)}} W_i, \quad D_i^{(2)} = T^{\varphi_i} V_i$$

$$D_x^{(3)} = g^{\alpha_2} T^{\sum\limits_{j=0}^{d} h_j x^j \zeta_x}, \quad D_x^{(4)} = T^{\zeta_x} R_{x'}.$$

We note that this sets $r_i = r\varphi_i$ and $r_x = r\zeta_x$. This is acceptable because $r_i$ is used as a modulo $p_1$ value and $\varphi_i$ is used as a modulo $p_2$ value and it is similarly with $r_x$. We set $\alpha_2' = 0$ for this is a nominally semi-functional key. Now the key is distributed as either a normal key or a nominally semi-functional key of type 1 depending on the value $T$.

If the $k$th key is in the revocation list $R$ in the challenger ciphertext, the key cannot decrypt the ciphertext whatever the key's type. If the $k$th key is not in the revocation list $R$ and the ciphertext satisfies the key's access policy, the key can decrypt the ciphertext whatever the key's type for the nominal mechanism. Next we argue that if the attacker dose not ask for a $k$th key that can decrypt the challenger ciphertext or the $k$th key that is not in the revocation list $R$ in the challenger ciphertext, the $k$th key is properly distributed in the attacker's view for a normal semi-functional key of type 1.

We assume that the $k$th key cannot decrypt the challenge ciphertext and be not in the revocation list $R$. This implies the rowspace $\boldsymbol{K}$ of the corresponding rows $i$ of matrix $\boldsymbol{A}$ that $\rho(i) \in \omega$ does not include the vector $\boldsymbol{l}$. Thus, we denote a vector $\boldsymbol{w}$ that $\boldsymbol{w}$ is orthogonal to $\boldsymbol{K}$, and not orthogonal to (1,0,$\cdots$,0). We set an equation that $\boldsymbol{v}_2 = f\boldsymbol{w} + \boldsymbol{v}_{2''}$ for $f \in \mathbf{Z}_N$ and $\boldsymbol{v}_{2''}$ is the span of the left rowspace. We note that $\boldsymbol{v}_{2''}$ reveals no information about $f$. For the rows $i$ that $\rho(i) \in \omega$, then in the subgroup $G_{p_2}$ we can only get $\boldsymbol{A}_i \cdot \boldsymbol{v}_{2''}$. We cannot get any information about $f\boldsymbol{w}$. For the rows $i$ that $\rho(i) \notin \omega$, we can get the $f\boldsymbol{w}$ in the equations: $\boldsymbol{A}_i \cdot \boldsymbol{v}_2 + \varphi_i s_{\rho(i)}$, where $\rho(i)$ are each unique attributes, as long as each $\varphi_i$ is not congruent to 0 modulo $p_2$. Each equations has a new unknown $s_{\rho(i)}$. In $D_x^{(3)}$ and $D_x^{(4)}$, $h_j$ modulo $p_2$ is unknown to the attacker that it cannot know whether $\alpha_{2'}$ is equal to 0. The value $\zeta_x$ modulo $p_2$ is from each other randomly. So we say that the value in the $G_{p_2}$ subgroup is information-theoretically hidden.

Therefore, $\mathcal{B}$ has properly simulated either $\text{Game}_{k-1,2}$ or $\text{Game}_{k,1}$ depending on the value $T$. Hence it can use the output of $\mathcal{A}$ to gain advantage negligibly close to $\epsilon$ in breaking Assumption 2.

**Lemma 3** Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $\text{Game}_{k,1}\text{Adv}_{\mathcal{A}} - \text{Game}_{k,2}\text{Adv}_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage $\epsilon$ in breaking Assumption 2.

**Proof** $\mathcal{B}$ is given as $\{g, X_3, g^s X_2, Y_2Y_3, T\}$. It will simulate either $\text{Game}_{k,1}$ or $\text{Game}_{k,2}$ with $\mathcal{A}$. It begins by sending $\mathcal{A}$ the public parameters $N, g, X_3, e(g,g)^\alpha, g^{s_i}, g_{h_j}$ where it chooses $\alpha$ and values $s_i, h_j$ randomly. In order to form the challenge ciphertext for a set of attribute $\omega$ and revocation list $R$, $\mathcal{B}$ sets

$$C = M_\beta e(g, g^s X_2)^\alpha, \quad C^{(1)} = g^s X_2,$$
$$C_i^{(2)} = (g^s X_2)^{s_i}\ (\forall i \in \omega),$$
$$C_x^{(3)} = (g^s X_2)^{\sum\limits_{j=0}^{d} h_j x^j}\ (\forall x \in \text{Cover}(R)).$$

We note that this implicitly sets $\xi_i = s_i$ and $\theta_x = \sum\limits_{j=0}^{d} h_j x^j \pmod N$. We say that values of $s_i$ and $h_j$ modulo $p_1$ are uncorrelated from the values of $s_i$ and $h_j$ modulo $p_2$. In order to form normal keys for queries greater than $k$, $\mathcal{B}$ can use MSK and do the regular key generation algorithm. To create semi-functional keys of type 2 for queries less than $k$, $\mathcal{B}$ first chooses random numbers $\alpha_1$ and $\alpha_2$ such that $\alpha = \alpha_1 + \alpha_2 \pmod N$. Then it chooses random vector $\boldsymbol{v}$ that $\boldsymbol{v} \cdot \boldsymbol{l} = \alpha_1$, random vector $\boldsymbol{v}_2'$, random number $r_i, r_x \in \mathbf{Z}_N$, random group elements $W_i, V_i, R_x$ and $R_x'$, and random number $\alpha' \in \mathbf{Z}_N$. The semi-functional key can be defined as

$$D_i^{(1)} = g^{\boldsymbol{A}_i \cdot \boldsymbol{v}} T_i^{r_i} W_i (Y_2Y_3)^{\boldsymbol{A}_i \cdot \boldsymbol{v}_2'}, \quad D_i^{(2)} = g^{r_i} V_i,$$
$$D_x^{(3)} = g^{\alpha_2} O^{r_x}(x) R_x (Y_2Y_3)^{\alpha'}, \quad D_x^{(4)} = g^{r_x} R_{x'}.$$

We note that for $Y_2 = g_2^c$, $\boldsymbol{\lambda}$ in our description of semi-functional keys above now corresponds to $\boldsymbol{\lambda} = c\boldsymbol{v}_2'$, $\alpha_2'$ corresponds to $\alpha_2' = c\alpha'$.

For the $k$th key for AS$(\boldsymbol{A}, \rho)$, $\mathcal{B}$ will make a key that is either a semi-functional key of type 1 or a semi-functional key of type 2 depending on the value of $T$ in the challenge. To form the $k$th key for AS$(\boldsymbol{A}, \rho)$, $\mathcal{B}$ first chooses random numbers $\alpha_1$ and $\alpha_2$ such that $\alpha = \alpha_1 + \alpha_2 \pmod N$. Then it chooses random vector $\boldsymbol{v}$ such that $\boldsymbol{v} \cdot \boldsymbol{l} = \alpha_1$ and random vector $\boldsymbol{v}_2$. Elements $\alpha_2', \varphi_i, \zeta_x, W_i$ and $V_i$ are chosen randomly.

$$D_i^{(1)} = g^{\boldsymbol{A}_i \cdot \boldsymbol{v}'} (Y_2Y_3)^{\boldsymbol{A}_i \cdot \boldsymbol{v}_2} T^{\varphi_i s_{\rho(i)}} W_i, \quad D_i^{(2)} = T^{\varphi_i} V_i,$$
$$D_x^{(3)} = g^{\alpha_2} (Y_2Y_3)^{\alpha_2'} T^{\sum\limits_{j=0}^{d} h_j x^j \zeta_x}, \quad D_x^{(4)} = T^{\zeta_x} R_{x'}.$$

If $T \in G$, this is a properly distributed semi-functional key of type 1. If $T \in G_{p_1 p_3}$, this is a properly distributed semi-functional key of type 2. Hence $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage $\epsilon$ in breaking Assumption 2.

**Lemma 4** Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $\text{Game}_{q,2}\text{Adv}_{\mathcal{A}} - \text{Game}_{\text{Final}}\text{Adv}_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage $\epsilon$ in breaking Assumption 3.

**Proof** $\mathcal{B}$ is given as $\{g, g^\alpha X_2, X_3, g^s Y_2, Z_2, T\}$. It will simulate $\text{Game}_{q,2}$ and $\text{Game}_{\text{Final}}$ with $\mathcal{A}$. It sets the public parameters as $N, g, X_3, e(g,g)^\alpha = e(g, g^\alpha X_2)$. And it chooses values of $s_i$ and $h_j$ randomly.

To form the challenge ciphertext for a set of attributes $\omega$ and revocation list $R$, $\mathcal{B}$ computes

$$C = M_\beta T, \quad C^{(1)} = g^s Y_2, \quad C_i^{(2)} = (g^s Y_2)^{s_i}\ (\forall i \in \omega),$$
$$C_x^{(3)} = (g^s Y_2)^{\sum\limits_{j=0}^{d} h_j x^j}\ (\forall x \in \text{Cover}(R)).$$

If $T = e(g,g)^{\alpha s}$, this will be a semi-functional ciphertext encryption of $M_\beta$. If $T$ is random, this will be a semi-functional ciphertext encryption of random message and will give no information about $\beta$ to attacker.

In order to form a semi-functional key of type 2 for AS$(\boldsymbol{A}, \rho)$ (for $n$ columns), $\mathcal{B}$ chooses $\alpha_2, v_2, \cdots, v_n$ randomly. Sets $\boldsymbol{v} = (\alpha - \alpha_2, v_2, \cdots, v_n)$($\alpha$ is from $g^\alpha X_2$). Then it chooses random $r_i, r_x \in \mathbf{Z}_N$, random group elements $W_i, V_i, R_x$ and $R'_x$, random vector $\boldsymbol{v}'$, and random $\alpha' \in \mathbf{Z}_N$. The semi-functional key can be defined as

$$D_i^{(1)} = g^{\sum\limits_{l=2}^{n} \boldsymbol{A}_{i,l} v_l} (g^\alpha X_2)^{\boldsymbol{A}_{i,1}} g^{-\alpha_2 \boldsymbol{A}_{i,1}} g^{s_i r_i} W_i (Z_2)^{\boldsymbol{A}_i \cdot \boldsymbol{v}'},$$

$$D_i^{(2)} = g^{r_i} V_i,$$

$$D_x^{(3)} = g^{\alpha_2} O^{r_x}(x) R_x (Z_2)^{\alpha'}, \quad D_x^{(4)} = g^{r_x} R_{x'}.$$

Then we have given a properly distributed semi-functional key of type 2. $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage $\epsilon$ in breaking Assumption 3.

After all, we can say that if Assumptions 1, 2 and 3 hold, we have shown by the previous lemmas that the real security game is indistinguishable from Game$_{\text{Final}}$, in which the value of $\beta$ is information theoretically hidden from the attacker. Hence the attacker cannot attain a non-negligible advantage in breaking our revocable ABE system.

## 6 Conclusion

We construct a fully secure key-policy ABE supporting revocation. By using the dual encryption system, we prove the security of our system in adaptive security model other than selective model which is much less practical. And we use complete binary tree to revoke the user in the leaf node.

## References

[1] SAHAI A, WATERS B. Fuzzy identity based encryption [C]// *24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Aarhus, Denmark: 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 [C]// *ACM Conference on Computer and Communications Security*. New York: ACM, 2006: 89-98.

[3] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption [C]// *IEEE Symposium on Security and Privacy*. Washington: IEEE Computer Society, 2007: 321-334.

[4] OSTROVSKY R, SAHAI A, WATERS B. Attribute based encryption with non-monotonic access structures [C]// *Proceedings of the 14th ACM Conference on Computer and Communications Security*. New York: ACM, 2007: 195-203.

[5] WATERS B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization [C]// *14th International Conference on Practice and Theory in Public Key Cryptography*. Taormina, Italy: Springer-Verlag, 2011: 53-70.

[6] BONEH D, FRANKLIN M. Identity based encryption from the weil pairing [C]// *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*. London: Springer-Verlag, 2001: 213-229.

[7] BOLDYREVA A, GOYAL V, KUMAR V. Identity-based encryption with efficient revocation [C]//*Proceedings of the 15th ACM Conference on Computer and Communications Security*. New York: ACM, 2008: 417-426.

[8] WATERS B. Dual system encryption: realizing fully secure ibe and hibe under simple assumptions [C]// *29th Annual International Cryptology Conference*. Santa Barbara: Springer-Verlag, 2009: 619-636.

[9] LEWKO A, WATERS B. New techniques for dual system encryption and fully secure hibe with short ciphertexts [C]// *7th Theory of Cryptography Conference*. Zurich, Switzerland: Springer-Verlag, 2010: 455-479.

[10] LEWKO A, OKAMOTO T, SAHAI A, et al. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption [C]// *29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. French Riviera: Springer-Verlag, 2010: 62-91.

[11] AIELLO W, LODHA S, OSTROVSKY R. Fast digital identity revocation (extended abstract) [C]// *18th Annual International Cryptology Conference Santa Barbara*. Santa Barbara: Springer-Verlag, 1998: 137-152.

[12] LIBERT B, VERGNAUD D. Adaptive-ID secure revocable identity-based encryption [C]// *The Cryptographers' Track at the RSA Conference 2009*. San Francisco: Springer-Verlag, 2009: 1-15.

[13] NAOR D, NAOR M, LOTSPIECH J. Revocation and tracing schemes for stateless receivers [C]//*21st Annual International Cryptology Conference*. Santa Barbara: Springer-Verlag, 2001: 41-62.