Directly Revocable and Verifiable Key-Policy Attribute-based Encryption for Large Universe

Hua Ma, Ting Peng, and Zhenhua Liu
(Corresponding author: Hua Ma)

School of Mathematics and Statistics, Xidian University
Xi'an 710071, P.R. China
(Email: hma@mail.xidian.edu.cn)

(Received Dec. 22, 2015; revised and accepted Apr. 9 & May 31, 2016)

Abstract

For practical data sharing applications, many attribute-based encryption (ABE) schemes were proposed with different kinds of properties, such as supporting large universe, revocation, verification and so on. However, existing schemes seldom support these three important properties simultaneously. In this paper, we present a directly revocable and verifiable key-policy ABE scheme for large universe (DRV-KP-ABE). The new scheme supports large universe, and attributes do not need to be enumerated at stage of setup. Meanwhile, our scheme allows the trusted authority to revoke users by only updating the revocation list without interaction with non-revoked users. We use the subset difference method for revocation which greatly improves the broadcast efficiency compared with the complete subtree scheme. In addition, the proposed scheme enables the third party to update ciphertexts with public information, and the auditor assures the third party updated ciphertexts correctly. The DRV-KP-ABE scheme is selectively secure under q-type assumption in the standard model.

Keywords: Attribute-based encryption, large universe, subset difference, user revocation, verification

1 Introduction

Recently, cloud computing has attracted wide attention from all walks of life. For one thing, cloud computing can provide powerful computing capabilities for resource-constrained devices. For another thing, cloud computing allows data users to store or deliver their sensitive data in third-party servers either for ease of sharing or for cost saving. However, there have some great challenges for preserving the privacy of stored data and enforcing access control on accessing these data [14, 28, 30, 32]. Attribute-based encryption (ABE) [26], introduced by Sahai and Waters, can be viewed as the right tool solving these challenges. In ABE, a party can encrypt a document to all users who have a certain set of attributes. For example, one can encrypt a document to all hiring committee members in the computer science department. In this case the document would be encrypted to the attribute subset {"Faculty", "CS Dept.", "Hiring Committee"}, and only users with all of these three attributes can hold the corresponding private keys and thus decrypt the document.

In practical data sharing system, if users leave or be removed from the system, their access right must be deprived. However, pure ABE scheme cannot revoke these users. In order to achieve revocation, revocable ABE (R-ABE) [1] was introduced. According to how to integrate the revocation information, existing R-ABE can be divided into two categories: directly R-ABE [29] and indirectly R-ABE [4]. Direct revocation enforces revocation by the sender who specifies the revocation list while encrypting data, and it is unnecessary for users to communicate with attribute authority. On the other hand, indirect revocation enforces revocation by the key authority who sends a key update information periodically to non-revoked users such that they can update own keys. The indirect method has an advantage that senders do not need to know the revocation list. However, this approach also has a disadvantage that the key update phase could be a bottleneck since the indirect revocation requires frequently communication between the key authority and all non-revoked users. In order to eliminate this bottleneck, we consider direct user revocation in this paper.

Till now, many R-ABE schemes [1, 2, 4, 15, 23, 27, 29] were proposed. Among these schemes, most of schemes essentially employ the complete subtree (CS) scheme [15] for revocation purpose. Replacing the CS method by the subset difference (SD) technique [9, 16] can reduce the size of the ciphertext component meant for performing revocation from $O(rlog\frac{n}{r})$ to $O(r)$, where n and r denote the number of users and the number of revoked users, respectively. This can provide significant improvement in the broadcast efficiency particularly when the number of users present in the system is very large compared with the number of revoked users [9].

In fact, R-ABE alone cannot prevent revoked users from decrypting ciphertexts that were generated before revocation, since the old private key of revoked users is enough to decrypt those ciphertexts. Thus a complete solution has to support not only the revocation functionality but also the ciphertext update functionality. Sahai and Waters [25] considered the problem of updating ciphertexts in the setting of R-ABE, where the third party (*e.g.*, storage server) can update stored ciphertexts with public available information. However, in a practical application setting [13, 27], if the correctness of updated ciphertexts cannot be guaranteed, the third party may return a wrong information for some interest. This motivates us to study ABE with verifiable ciphertext delegation.

In addition, ABE can be classified into two categories depending on the size of attributes: small universe construction [3] and large universe construction [24]. In "small universe" construction, attributes are pre-specified at the stage of setup, and additional attributes cannot be added. While in "large universe" construction, the size of the attributes can be exponentially large, and a large number of new attributes can be added to the system at any time. Therefore, it seems that the scheme supporting large universe is more suitable for actual demand.

1.1 Our Contribution

We present a KP-ABE scheme that simultaneously supports user revocation, ciphertext update verification and "truly" large attribute universe. For that purpose, we make use of the technique given by Rouselakis and Waters [24] to achieve large universe construction and the one given by Shi et al. [27] in order to achieve the revocation and verification. Specially, we manage revocation for ABE by utilizing the SD mechanism [9, 16] instead of CS scheme. More precisely, we present the KP-ABE scheme with the following properties:

- The proposed scheme has a constant number of group elements in public parameters, and imposes no bound on the size of attributes used for encryption.
- In our scheme, the private key is associated with a user identity and the access policy. The ciphertext is related to the set of attributes and a revocation list. A user can decrypt a ciphertext if and only if her/his attributes satisfy the access structure and she/he is not in the revocation list. When revoking users, the trusted authority only needs to update the revocation list, without any interaction with non-revoked users.
- We use the SD method for revocation which greatly improves the broadcast efficiency and provides a smaller covering set compared with the CS scheme.
- Our scheme can delegate the third party to update ciphertexts with public available information. Meanwhile, the scheme allows the auditor to verify whether ciphertexts are updated correctly by the third party.

1.2 Organization

The rest of the paper is organized as follows. We discuss related works in Section 2 and in Section 3 we give necessary background information. We describe scheme definition and security game in Section 4. The concrete construction of DRV-KP-ABE scheme and security proof are detailed in Sections 5 and 6, respectively. The performance analysis and efficiency improvement are discussed in Section 7. Finally, we give the conclusion in Section 8.

2 Related Works

The notion of ABE, which enables fine-grained and non-interactive access control of shared data, was first introduced by Sahai and Waters [26]. Currently, there are two different and complementary forms of ABE: key-policy ABE (KP-ABE) [19] and ciphertext-policy ABE (CP-ABE) [3, 10, 21]. In KP-ABE, the ciphertext is associated with the set of attributes, and the secret key is associated with the access policy. While in CP-ABE, the idea is reversed: the user' private key is related to the set of attributes, and the access control policy on these attributes is attached to the ciphertext. Until now, many variants of ABE have been proposed to provide promising properties and functionalities, such as multi-authority ABE [18], ABE with outsourcing decryption [13, 36], traceable ABE [22], anonymous ABE [34] and so on. As a promising property, R-ABE is still an active research topic.

In 2008, Boldyreva et al. [4] constructed the first R-ABE scheme with indirect revocation, which key authority sends a key update information periodically to non-revoked users. Meanwhile, the sender doesn't care the revocation list when encrypting a message (*e.g.*, [4, 15]). Attrapadung and Imai [1] proposed a directly revocable ABE scheme, which the trusted authority revokes users by only updating the revocation list without interaction with non-revoked users. Later, they proposed a hybrid R-ABE scheme [2] which allows senders to choose the concrete revocation mode during the encryption phase. However, all of the above constructions make use of the CS method to achieve revocation. Lee et al. [16] utilized the SD scheme to achieve revocation for identity-based encryption (IBE) and pointed out that their technique for R-IBE cannot be directly applicable to construct an R-ABE scheme. Recently, Datta et al. [9] presented the first fully secure unbound R-ABE scheme in prime order bilinear groups via SD mechanism.

Note that these mechanisms in the above revocation modes only guarantee that revoked users cannot decrypt ciphertexts created after revocation. To prevent revoked users from decrypting ciphertexts that were generated before revocation, proxy re-encryption [8, 23] was introduced, which needs the interaction between the proxy and the trusted authority. Other works [25, 27, 33] considered the problem of updating ciphertexts in the setting of R-ABE, where the third party can update stored cipher-

texts without any interaction with either data owners or the trusted authority whenever the revocation event happens. In addition, Shi et al. [27] considered ciphertext update verification in the setting of directly R-ABE, where the auditor can verify whether ciphertexts were updated correctly or not.

Though these schemes have been introduced to achieve efficient revocation, they cannot support large attribute universe construction. The first large universe KP-ABE scheme in the standard model was given in [17], which is based on composite order groups. Later, several large universe ABE schemes were given in [12, 15, 24]. However, they have no consider the revocation and verification, simultaneously. Thus, it is necessary to study revocable and verifiable KP-ABE scheme for large universe.

3 Preliminaries

In this section, we give the definitions of bilinear groups, access structures and complexity assumptions. In addition, in order to achieve efficient revocation, we introduce the binary tree and SD method.

3.1 Notations

For $n \in \mathbb{N}$, we define $[n] = \{1, 2, \cdots, n\}$. Similarly, for $n_1, \cdots, n_k \in \mathbb{N} : [n_1, \cdots, n_k] = [n_1] \times [n_2] \times \cdots \times [n_k]$. When S is a set, we denote by $s \leftarrow S$ the fact that the variable s is picked uniformly at random from S. We write $s_1, s_2, \cdots, s_n \leftarrow S$ as shorthand for $s_1 \leftarrow S, s_2 \leftarrow S, \cdots, s_n \leftarrow S$. When v is a vector (of any type), we will denote by v_i the i-th element and by $\langle v, w \rangle$ the inner product of vectors v and w.

3.2 Bilinear Groups

Let $\mathbb{G}$, $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}$. A bilinear map e is a map $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ with the following properties:

1) **Bilinearity**: For all $u, v \in \mathbb{G}, a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;

2) **Non-degeneracy**: $e(g, g) \neq 1$;

3) **Computability**: There is an efficient algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}$.

3.3 Access Structures

In this section, we present the formal definition of access structures [24]. Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ be the attribute universe. An access structure on $\mathcal{P}$ is a collection $\mathbb{A}$ of non-empty sets of attributes, *i.e.*, $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called the unauthorized sets. Additionally, an access structure is called monotone if $\forall B, C \in \mathbb{A}$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. In our construction, we only consider monotone access structures.

3.4 Linear Secret-Sharing Schemes

A linear secret-sharing scheme (LSSS) [24] can be used to represent an access control policy (M, ρ), where M is an $l \times n$ matrix which is called the share generating matrix and ρ maps a row into an attribute. A LSSS consists of two algorithms:

Share$((M, \rho), s)$**:** This algorithm is used to share secret value s to attributes. Considering a column vector $v = (s, r_2, \cdots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \cdots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\lambda_i = M_i \cdot v$ is a share of the secret s, which belongs to the attribute $\rho(i)$.

Reconstruction$(\lambda_1, \cdots, \lambda_l, (M, \rho))$**:** This algorithm is used to reconstruct s from secret shares. Let $S \in \mathbb{A}$ be any authorized set and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \cdots, l\}$. Then there exists coefficients $\{c_i | i \in I\}$ such that $\Sigma_{i \in I} c_i M_i = (1, 0, \cdots, 0)$, thus we have $\Sigma_{i \in I} c_i \lambda_i = s$.

3.5 Subset Difference Method

In this paper, we use the subset difference method [9, 16] to realize efficient revocation. We now give some notations (as shown in Table 1) concerning a full binary tree which is similar to those defined in literature [9]. Label the nodes in $\mathcal{BT}$ by 1 to $2n-1$ in the way that root is labeled 1, if parent is labeled i, then the left child is labeled $2i$ and the right child is labeled $2i+1$. Every member in U is assigned to a leaf node in the tree. The identifier L_i of each node in the tree is assigned as follows: Each edge in the tree is assigned with 0 or 1 depending on whether the edge connects a node to its left or right child node. The identifier L_i of a node v_i is the bit string obtained by reading all the labels of edges in the path from the root to the node v_i. The Steiner Tree $ST(R)$ is the minimal subtree of $\mathcal{BT}$ that connects all the leaf nodes in R and the root node. One example of the labeling is shown in Figure 1.

The SD scheme [9] is summarized as follows:

SD.Setup(n)**:** This algorithm takes as input the maximum number n of users (for simplicity, $n = 2^d$). It sets a full binary tree $\mathcal{BT}$ of depth d and assigns every user to a different leaf node in $\mathcal{BT}$. It outputs $\mathcal{BT}$.

SD.Assign$(\mathcal{BT}, u)$**:** This algorithm takes as input the full binary tree $\mathcal{BT}$ and a user serial number $u \in [n]$. Let $v(u)$ be the leaf node assigned to u and $path(v(u)) = (v_{k_0}, v_{k_1}, ..., v_{k_n})$ be the path from the root node v_{k_0} to the leaf node $v_{k_n} = v(u)$. For all $i, j \in \{k_0, k_1, ..., k_n\}$ such that v_j is a descendant of v_i, it adds $S_{i,j}$ defined by two nodes v_i and v_j in the path into a private set PV_u. Finally, it outputs PV_u.

SD.Cover$(\mathcal{BT}, R)$**:** This algorithm takes as input the binary tree $\mathcal{BT}$ and a revoked set R of users. Then it

Table 1: Notations

Notation	Significance
n	The maximum number of users in the system
$\mathcal{BT}$	A full binary tree with n leaf nodes
U	A serial number set of user
R	A serial number set of revoked user
v_i	A node in $\mathcal{BT}$ for any $i, i \in [2n-1]$
d_i	The depth of the node v_i
T_i	A subtree of $\mathcal{BT}$ that is rooted at v_i, where $v_i \in \mathcal{BT}$
$T_{i,j}$	A subtree $T_i - T_j$ for any $v_i, v_j \in \mathcal{BT}$ such that v_i is the ancestor of v_j
S_i	The set of leaf nodes of T_i
$S_{i,j}$	The set of leaf nodes of $T_{i,j}$
L_i	An identifier for a node v_i in $\mathcal{BT}$, a fixed and unique string
$(L_i\|\|d_j)$	The integer representation of the string formed by concatenating d_j with L_i
$(L_i\|\|L_j)$	The integer representation of the string obtained by concatenating L_j with L_i
$ST(R)$	The Steiner Tree induced by a subset R and the root node

computes CV_R iteratively by removing nodes from $ST(R)$ until $ST(R)$ only has a single node as follows (an example is given in Figure 1):

1) Find two leaves v_i and v_j in $ST(R)$ such that the least-common-ancestor (lca) v of v_i and v_j does not contain any other leaf node of $ST(R)$ in its subtree. Let v_l and v_k be the two children of v such that v_i is a descendant of v_l, and v_j is a descendant of v_k. (If there is only one leaf node, make $v_i = v_j$ to be the leaf, v to be the root of $ST(R)$ and $v_l = v_k = v$.)
2) If $v_l \neq v_i$, $CV_R = CV_R \bigcup S_{l,i}$; if $v_k \neq v_j$, $CV_R = CV_R \bigcup S_{k,j}$.
3) Remove from $ST(R)$ all the descendants of v and make it a leaf.

SD.Match(CV_R, PV_u)**:** This algorithm takes as input a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_u = \{S_{i',j'}\}$. It obtains $(S_{i,j}, S_{i',j'})$ such that $S_{i,j} \in CV_R, u \in S_{i,j}$, and $S_{i',j'} \in PV_u$, or obtains $\perp$.

For all $n = 2^d$, let $\mathcal{BT} \leftarrow$ **SD.Setup**(n), $PV_u \leftarrow$ **SD.Assign**$(\mathcal{BT}, u)$, $CV_R \leftarrow$ **SD.Cover**$(\mathcal{BT}, R)$. The correctness of the SD scheme is defined as follows:

1) If $u \notin R$, then **SD.Match**$(CV_R, PV_u) = (S_{i,j}, S_{i',j'})$;
2) If $u \in R$, then **SD.Match**$(CV_R, PV_u) = \perp$.

As shown in Figure 1, given $R = \{u_2, u_3, u_4, u_6\}$. According to the algorithm **SD.Cover**$(\mathcal{BT}, R)$, choose v_5 as a lca of v_{10} and v_{11}. Get v_5 as a leaf, v_2 is a lca of v_9 and v_5. Get v_2 as a leaf, $CV_R = \{S_{4,9}\}$, the root is a lca of v_{13} and v_2, $CV_R = \{S_{4,9}, S_{3,13}\}$. For $u_1 \notin R$, $PV_{u_1} = \{S_{1,2}, S_{1,4}, S_{1,8}, S_{2,4}, S_{2,8}, S_{4,8}\}$, run the algorithm **SD.Match**(CV_R, PV_{u_1}) and output $(S_{4,9}, S_{4,8})$. Similarly, the algorithm **SD.Match**(CV_R, PV_{u_8}) outputs $(S_{3,13}, S_{3,15})$. For $u \in R$, the algorithm **SD.Match**(CV_R, PV_u) outputs $\perp$.

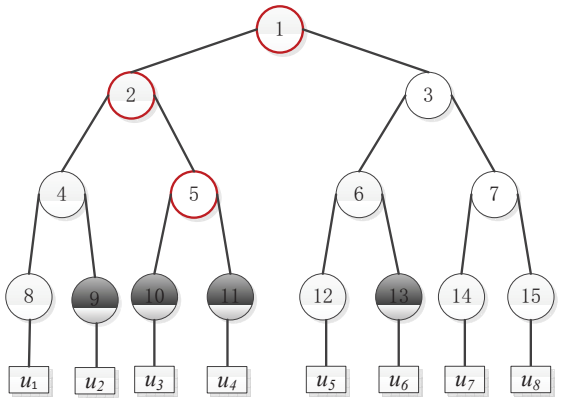

Figure 1: An Example of Binary Tree $\mathcal{BT}$

Lemma 1. *Let n be the number of leaf nodes and r be the size of a revoked set. The size of a private set is $O(log^2 n)$ and the size of a covering set is at most $2r-1$ in the SD scheme [9].*

3.6 Complexity Assumption

For our KP-ABE construction we will use a q-type assumption [24], which is similar to the Decisional Bilinear Diffie-Hellman Assumption augmented with q parameters b_i. The assumption is defined via the following game between a challenger and an attacker:

Initially, the challenger inputs the security parameter and picks a random group element $g \leftarrow \mathbb{G}$ and $q+3$ random exponents $a, b, c, b_1, b_2, \cdots, b_q \leftarrow \mathbb{Z}_p$. Then he sends the group description $(p, \mathbb{G}, \mathbb{G}_T, e)$ and all of the

following terms to the attacker:

$$g, g^a, g^b, g^c, g^{(ac)^2},$$
$$g^{b_i}, g^{acb_i}, g^{ac/b_i}, g^{a^2cb_i}, g^{b/b_i^2}, g^{b^2/b_i^2}, \forall i \in [q],$$
$$g^{acb_i/b_j}, g^{bb_i/b_j^2}, g^{abcb_i/b_j}, g^{(ac)^2b_i/b_j}, \forall i, j \in [q], i \neq j.$$

The challenger flips a random coin $\mu \leftarrow \{0,1\}$ and if $\mu = 0$ he gives the term $e(g,g)^{abc}$ to the attacker. Otherwise he gives a random term $\mathcal{R} \leftarrow \mathbb{G}_T$. Finally the attacker outputs a guess $\mu' \in \{0,1\}$.

Definition 1. *We say that the q-type assumption holds if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage is defined as*

$$Adv = |\Pr[\mu' = \mu] - 1/2|.$$

4 Syntax and Security

4.1 Algorithms

A directly revocable and verifiable key-policy attribute-based encryption (KP-ABE) scheme for large universe consists of the following six algorithms:

Setup$(1^\lambda, n) \rightarrow (pp, msk)$**:** Input a security parameter λ and the maximum number n of users. The algorithm obtains $\mathcal{BT}$ by running **SD.Setup**(n) and outputs the public parameters pp and the master secret key msk. We assume that the public parameters contain a description of the attribute universe $\mathcal{N}$.

Extract$(msk, ID, \mathbb{A}) \rightarrow sk$**:** Given a user identity ID, choose an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assign ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. (For convenience, a serial number and the corresponding user identity can be exchanged in this paper.) The algorithm runs **SD.Assign**$(\mathcal{BT}, u)$ to obtain $PV_u = \{S_{i,j}\}$. Input the master secret key msk and an access structure $\mathbb{A}$ on $\mathcal{N}$. The algorithm generates a secret key corresponding to $\mathbb{A}$ and ID.

Encrypt$(pp, m, S, R) \rightarrow ct$**:** Input the public parameters pp, a plaintext message m, and a set of attributes $S \subseteq \mathcal{N}$, as well as the current revocation list R. The algorithm obtains the cover set CV_R by executing **SD.Cover**$(\mathcal{BT}, R)$ and outputs the ciphertext ct.

Decrypt$(sk, ct) \rightarrow m$**:** Input a secret key sk, and a ciphertext ct, if the attribute set S satisfies the ciphertext policy and user identity $ID \notin R$, then user obtains $(S_{i,j}, S_{\tilde{i},\tilde{j}})$ by running **SD.Match**(CV_R, PV_u) such that $S_{i,j} \in CV_R, S_{\tilde{i},\tilde{j}} \in PV_u$ and $(i = \tilde{i}) \wedge (d_j = d_{\tilde{j}}) \wedge (j \neq \tilde{j})$ and recovers a message m. Otherwise, return $\perp$.

CTUpdate$(ct, R', pp) \rightarrow ct'$**:** Input ct, the latest revocation list R' and pp, the third party updates the original ciphertext ct to ct' associated with R'.

Verify$(ct, ct') \rightarrow \{0,1\}$**:** After the third party finished the ciphertexts updating, the auditors will verify whether ciphertexts were updated correctly from some prior revocation lists to the current revocation lists or not. Then he outputs 1 if the update is correct, and 0 otherwise.

Let $(pp, msk) \leftarrow$ **Setup**(1^λ), $ct \leftarrow$ **Encrypt**(pp, m, S, R), $ct' \leftarrow$ **CTUpdate**(ct, R', pp), $sk \leftarrow$ **Extract**$(msk, ID, \mathbb{A})$. For correctness, we require the following conditions always hold:

1) $1 \leftarrow$ **Verify**(ct, ct').
2) If $ID \notin R$ and the attribute set S satisfies the ciphertext policy $\mathbb{A}$, the algorithm returns $m \leftarrow$ **Decrypt**(sk, ct). Otherwise, return $\perp$.
3) If $ID \notin R'$ and the attribute set S satisfies the ciphertext policy $\mathbb{A}$, the algorithm returns $m \leftarrow$ **Decrypt**(sk, ct'). Otherwise, return $\perp$.

4.2 Selective Security Game

Similar to the security model in the literature [27], since the updated ciphertexts have the same distribution as original ciphertexts, we only consider the security of original ciphertexts. We now describe the security model of our system by the following game between a challenger and an attacker:

Initialization: In this phase, the attacker $\mathcal{A}$ declares the challenge attribute set S^* and a revocation list R^*, which he will try to attack, and sends them to the challenger $\mathcal{B}$.

Setup: The challenger runs **Setup**(1^λ) to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$.

Query phase 1: In this phase, the attacker $\mathcal{A}$ can adaptively issue extract queries for secret keys related to several tuples $(ID, \mathbb{A})$.

- If $S \in \mathbb{A}$ and $ID \notin R^*$, then $\mathcal{B}$ will abort.
- Otherwise, the challenger generates a secret key related to $(ID, \mathbb{A})$ for the attacker $\mathcal{A}$.

Challenge: The attacker $\mathcal{A}$ declares two equal-length plaintexts m_0 and m_1 and submits them to the challenger. The challenger flips a random coin $\nu \in \{0,1\}$ and calls **Encrypt**$(m_\nu, S^*, R^*) \rightarrow ct$. He sends ct to the attacker.

Query phase 2: Phase 2 is the same as Phase 1.

Guess: The attacker $\mathcal{A}$ outputs his guess $\nu' \in \{0,1\}$ for ν.

Definition 2. *A KP-ABE scheme is selectively secure if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage of an attacker is defined as*

$$Adv = |\Pr[\nu = \nu'] - 1/2|.$$

4.3 Verifiability Game

In this subsection, we give the security model of verifiability based on the literature [27, 36].

Initialization: In this phase, the attacker $\mathcal{A}$ chooses an attribute set S^* and sends it to the challenger.

Setup: The challenger runs $\textbf{Setup}(1^\lambda)$ to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$.

Query phase 1: In this phase, the attacker $\mathcal{A}$ can adaptively issue queries:

1) **Extract query**: Input several tuples $(ID, \mathbb{A})$, the challenger generates a secret key sk related to $(ID, \mathbb{A})$ for the attacker $\mathcal{A}$.
2) **Verification query**: Input updated ciphertexts ct^*, the challenger returns γ to $\mathcal{A}$ by running $\textbf{Verify}(ct, ct^*) \to \gamma$.

Challenge: The attacker $\mathcal{A}$ selects a plaintext m and two revocation lists R and R^*, where $R \subset R^*$, and submits them to the challenger. The challenger calls $\textbf{Encrypt}(m, S^*, R) \to ct$. He sends ct to the attacker.

Guess: The attacker $\mathcal{A}$ outputs a updated ciphertext ct^* to the challenger associated with the revocation list R^*. The attacker $\mathcal{A}$ wins this game if $\textbf{Verify}(ct, ct^*) \to 1$ and the distribution of ct^* and ct' are computationally distinguishable, where $\textbf{Update}(ct, R^*, pp) \to ct'$ produced by the challenger.

Definition 3. *We say that the proposed scheme achieves update verifiability if the advantage that any $\mathcal{A}$ wins the verifiability game is negligible in security parameter λ.*

5 Our Construction

We construct a directly revocable and verifiable KP-ABE scheme for large universe based on the techniques in paper [24, 27]. Different from the scheme in the literature [27], we use the SD scheme to manage revocation for ABE. The SD method provides a smaller covering set compared with the CS scheme, particularly when the number of users present in the system is very large. In addition, we also use the method given by Lee et al. [16] to solve the complex key assignment problem of the SD mechanism. Now we describe the scheme as follows.

$\textbf{Setup}(1^\lambda, n) \to (pp, msk)$: Input a security parameter λ and the maximum number n of users. The algorithm obtains $\mathcal{BT}$ by running $\textbf{SD.Setup}(n)$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. The attribute universe is $\mathcal{N} = \mathbb{Z}_p$. Select the random terms $g, z, h, w, f \leftarrow \mathbb{G}$ and $\alpha, \beta \leftarrow \mathbb{Z}_p$. Let $v = g^\beta$. The public parameters are published as

$$pp = (g, z, h, w, v, f, e(g, g)^\alpha).$$

The authority sets $msk = (\alpha, \beta)$ as the master secret key.

$\textbf{Extract}(msk, ID, (M, \rho)) \to sk$: The algorithm picks $y = (\alpha_1, y_2, \cdots, y_n)^\top$ and sets α_2 such that $\alpha = \alpha_1 + \alpha_2 (\text{mod } p)$, where $\alpha_1, y_2, \cdots, y_n \leftarrow \mathbb{Z}_p$ and α_1 is the secret to be shared among the shares. The vector of the shares is

$$M \cdot y = (\lambda_1, \lambda_2, \cdots, \lambda_l)^\top.$$

Select l random exponents $k_1, k_2, \cdots, k_l \leftarrow \mathbb{Z}_p$ and for every $\tau \in [l]$

$$K_{\tau,0} = g^{\lambda_\tau} w^{k_\tau}, K_{\tau,1} = (z^{\rho(\tau)} h)^{-k_\tau}, K_{\tau,2} = g^{k_\tau}.$$

Given user identity ID, choose an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assign ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. Next the algorithm obtains PV_u by running $\textbf{SD.Assign}(\mathcal{BT}, u)$. For each $S_{i,j} \in PV_u$, pick $\theta_{i,j,1}, \theta_{i,j,2} \leftarrow \mathbb{Z}_p$ and compute

$$D_{i,j,1} = g^{\alpha_2} v^{(L_i || d_j)\theta_{i,j,1}} f^{\theta_{i,j,2}}, E_{i,j,1} = g^{-\theta_{i,j,1}},$$
$$D_{i,j,2} = (f^{(L_i || L_j)} v)^{\theta_{i,j,2}}, E_{i,j,2} = g^{-\theta_{i,j,2}}.$$

The secret key is

$$sk = (PV_u, ID, (M, \rho), \{K_{\tau,0}, K_{\tau,1}, K_{\tau,2}\}_{\tau \in [l]}, \{D_{i,j,t}, E_{i,j,t}\}_{S_{i,j} \in PV_u, t=1,2}).$$

$\textbf{Encrypt}(m, S = \{A_1, A_2, \cdots, A_k\} \subseteq Z_p, R, pp) \to ct$: The algorithm picks $k+1$ random exponents $s, r_1, r_2, \cdots, r_k \leftarrow \mathbb{Z}_p$ and computes

$$C = m \cdot e(g, g)^{\alpha s}, C_0 = g^s,$$
$$\forall \tau \in [k], C_{\tau,1} = g^{r_\tau}, C_{\tau,2} = (z^{A_\tau} h)^{r_\tau} w^{-s}.$$

Given the revoked user serial number set $R \subseteq [n]$, the algorithm obtains the cover set CV_R by executing $\textbf{SD.Cover}(\mathcal{BT}, R)$. For each $S_{i,j} \in CV_R$, set

$$C_{i,j,1} = v^{(L_i || d_j) s}, C_{i,j,2} = (f^{(L_i || L_j)} v)^s.$$

The ciphertext is

$$ct = (S, R, CV_R, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau \in [k]}, \{C_{i,j,t}\}_{S_{i,j} \in CV_R, t=1,2}).$$

$\textbf{Decrypt}(sk, ct) \to m$: Given sk and ct, the decryption can be done as follows:

- If user identity $ID \in R$ or the attribute set S does not satisfy the ciphertext policy, then return $\perp$.
- Otherwise, proceed as follows. Suppose the set S satisfies the access structure (M, ρ) and $ID \notin R$. Let $I = \{i : \rho(i) \in S\}$. There exists constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i M_i =$

$(1, 0, \cdots, 0)$. Since $ID \notin R$, then user obtains $(S_{i,j}, S_{\tilde{i},\tilde{j}})$ by running **SD.Match**(CV_R, PV_u) such that $S_{i,j} \in CV_R, S_{\tilde{i},\tilde{j}} \in PV_u$ and $(i = \tilde{i}) \wedge (d_j = d_{\tilde{j}}) \wedge (j \neq \tilde{j})$.
Then the user calculates

$$\begin{aligned} B =& \prod_{i \in I} (e(C_0, K_{i,0}) e(C_{\tau,1}, K_{i,1}) \\ & \cdot e(C_{\tau,2}, K_{i,2}))^{\omega_i}, \\ B' =& e(C_0, D_{\tilde{i},\tilde{j},1}) e(C_{i,j,1}, E_{\tilde{i},\tilde{j},1}) [e(C_0, D_{\tilde{i},\tilde{j},2}) \\ & \cdot e(C_{i,j,2}, E_{\tilde{i},\tilde{j},2})]^{\frac{-1}{(L_{\tilde{i}}||L_{\tilde{j}}) - (L_i||L_j)}}, \\ m =& C/(BB'), \end{aligned}$$

where τ is the index of the attribute $\rho(i)$ in S (it depends on i).

CTUpdate$(ct, R', pp) \rightarrow ct'$**:** Given the latest revoked user serial number set R', the cover set $CV_{R'}$ is obtained by running **SD.Cover**$(\mathcal{BT}, R')$. For $S_{i',j'} \in CV_{R'}$, there are two cases.

Case 1: If there exists $S_{i,j} \in CV_R$ such that $(i = i') \wedge (j = j')$ then set $ct' = ct$.

Case 2: Otherwise, there exists $S_{i,j} \in CV_R$ such that v_i is an ancestor of $v_{i'}$ or $v_i = v_{i'}$. Choose the random exponent $s' \leftarrow \mathbb{Z}_p$ and set

$$\begin{aligned} C_{i',j',1} =& (C_{i,j,1})^{\frac{(L_{i'}||d_{j'})}{(L_i||d_j)}} \cdot v^{(L_{i'}||d_{j'})s'}, \\ C_{i',j',2} =& (C_{i,j,2})^{\frac{(L_{i'}||L_{j'})}{(L_i||L_j)}} \cdot f^{(L_{i'}||L_{j'})s'} \\ & (C_{i,j,1})^{\frac{1}{(L_i||d_j)}(1 - \frac{(L_{i'}||L_{j'})}{(L_i||L_j)})} \cdot v^{s'}. \end{aligned}$$

Finally, let $C' = C \cdot e(g,g)^{\alpha s'}, C'_0 = C_0 \cdot g^{s'}, C'_{\tau,1} = C_{\tau,1}, C'_{\tau,2} = C_{\tau,2} \cdot w^{-s'}, \forall \tau \in [k]$. The updated ciphertext is

$$\begin{aligned} ct' =& (S, R', CV_{R'}, C', C'_0, \{C'_{\tau,1}, C'_{\tau,2}\}_{\tau \in [k]}, \\ & \{C'_{i',j',t}\}_{S_{i',j'} \in CV_{R'}, t=1,2}). \end{aligned}$$

Verify$(ct, ct') \rightarrow \{0, 1\}$**:** The verification can be done as follows:

- Verify whether the following equation holds:

$$\begin{aligned} & e(g, C'_{\tau,2} C'/C) e(C'_0, w) \\ & = e(C'_{\tau,1}, z^{A_\tau} h) e(C'_0/C_0, e(g,g)^{\alpha}), \end{aligned}$$

where $\tau \in [k]$.
If not, then output 0. Otherwise, proceed to the following step.

- Figure out $S_{i'_1,j'_1}, \cdots, S_{i'_\eta,j'_\eta}$ such that $S_{i'_k,j'_k} \in CV_{R'} - CV_R$, where $k \in \{1, \cdots, \eta\}$, select $c_1, \cdots, c_\eta \leftarrow \mathbb{Z}_p$ and verify

$$\begin{aligned} & e(C'_0, \prod_{k=1}^{\eta} (v^{(L_{i'_k}||d_{j'_k})})^{c_k}) = e(g, \prod_{k=1}^{\eta} (C'_{i'_k,j'_k,1})^{c_k}), \\ & e(C'_0, \prod_{k=1}^{\eta} (f^{(L_{i'_k}||L_{j'_k})} v)^{c_k}) = e(g, \prod_{k=1}^{\eta} (C'_{i'_k,j'_k,2})^{c_k}). \end{aligned}$$

If the above equations do not hold, then return 0. Otherwise, return 1.

6 Security Proof

Since the updated ciphertexts has the same distribution as the original ciphertexts. We only prove the security related to the original ciphertexts and updated verifiability.

Theorem 1. *If the q-type assumption holds, then all PPT attackers with a challenge attribute set of size k, where $k \leq q$, have a negligible advantage in selectively breaking our scheme.*

Proof. To prove the theorem we will assume that there exists a PPT attacker $\mathcal{A}$ with a challenge attribute set, which has a non-negligible advantage $Adv_{\mathcal{A}}$ in selectively breaking our scheme. Using this attacker we will build a PPT challenger $\mathcal{B}$ that attacks the q-type assumption with a non-negligible advantage.

Initialization: Initially, $\mathcal{B}$ receives the given terms from the assumption, an attribute set $S^* = \{A_1^*, A_2^*, \cdots, A_k^*\} \subseteq N$ and a revocation list R^*.

Setup: Now, the challenger $\mathcal{B}$ provides the public parameters of the system for $\mathcal{A}$. In order to do that $\mathcal{B}$ implicitly sets the master secret key of the scheme to be $\alpha = ab$ and $\beta = b + d$, where a, b are set in the assumption and $d \leftarrow \mathbb{Z}_p$. $\mathcal{B}$ picks the random terms $f, \tilde{z}, \tilde{h} \leftarrow \mathbb{Z}_p$ and gives the following terms to $\mathcal{A}$:

$$\begin{aligned} g &= g, \\ z &= g^{\tilde{z}} \cdot \prod_{i \in [k]} g^{b/b_i^2}, \\ h &= g^{\tilde{h}} \cdot \prod_{i \in [k]} g^{ac/b_i} \cdot \prod_{i \in [k]} (g^{b/b_i^2})^{-A_i^*}, \\ w &= g^a, \\ v &= g^{\beta} = g^{b+d}, \\ e(g,g)^{\alpha} &= e(g^a, g^b). \end{aligned}$$

Phase 1: In this phase, the attacker $\mathcal{A}$ issues extract queries for secret keys related to several tuples $(ID, (M, \rho))$, and the challenger proceeds as follows:

Case 1: If $S^* \in (M, \rho)$ and $ID \notin R^*$, then $\mathcal{B}$ will abort.

Case 2: If $ID \in R^*$, the challenger selects $\alpha_1 \in \mathbb{Z}_p$ and computes $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ based on the algorithm **Extract**$(msk, (M, \rho))$. Next the challenger chooses an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assigns ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. Then he obtains PV_u by running **SD.Assign**$(\mathcal{BT}, u)$. For each $S_{i,j} \in PV_u$,

choose $\theta'_{i,j,1}, \theta_{i,j,2} \in \mathbb{Z}_p$ at random and set

$$\begin{aligned}
D_{i,j,1} =& g^{-\alpha_1} g^{b(L_i||d_j)\theta'_{i,j,1}} f^{\theta_{i,j,2}} \\
& \cdot g^{d(L_i||d_j)(\theta'_{i,j,1} - \frac{a}{(L_i||d_j)})}, \\
E_{i,j,1} =& g^{-\theta_{i,j,1}} = g^{\frac{a}{(L_i||d_j)} - \theta'_{i,j,1}}, \\
D_{i,j,2} =& (f^{(L_i||L_j)} v)^{\theta_{i,j,2}}, \\
E_{i,j,2} =& g^{-\theta_{i,j,2}}
\end{aligned}$$

by implicitly defining

$$\begin{aligned}
\alpha_2 &= \alpha - \alpha_1 = ab - \alpha_1, \\
\theta_{i,j,1} &= \theta'_{i,j,1} - \frac{a}{(L_i||d_j)}.
\end{aligned}$$

Case 3: If $S^* \notin (M, \rho)$, the challenger selects $\alpha_2 \in \mathbb{Z}_p$ and computes $D_{i,j,1}, E_{i,j,1}, D_{i,j,2}, E_{i,j,2}$ based on the algorithm **Extract**(msk, ID). Since $S^* \notin (M, \rho)$, there exists a vector $\omega = (\omega_1, \omega_2, \cdots, \omega_n)^\top \in \mathbb{Z}_p^n$ such that $\omega_1 = 1$ and $\langle M_\tau, \omega \rangle = 0$ for all $\tau \in [l]$ such that $\rho(\tau) \in S^*$. The vector y that will be shared is implicitly

$$y = (\alpha - \alpha_2)\omega + (0, \tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n)^\top,$$

where $\tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n \leftarrow \mathbb{Z}_p$. For each row $\tau \in [l]$ the share is

$$\lambda_\tau = \langle M_\tau, y \rangle = (ab - \alpha_2)\langle M_\tau, \omega \rangle + \tilde{\lambda}_\tau.$$

Then the challenger computes $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ as follows:

- If $\rho(\tau) \in S^*$: In this case $\lambda_\tau = \langle M_\tau, y \rangle = \tilde{\lambda}_\tau = \langle M_\tau, (0, \tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n)^\top \rangle$; hence its value is known to the challenger. Pick $K_\tau \in \mathbb{Z}_p$ and output the terms $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ as in the algorithm **Extract**.
- Otherwise $\rho(\tau) \notin S^*$: Pick $\tilde{k}_\tau \in \mathbb{Z}_p$ and compute

$$\begin{aligned}
K_{\tau,0} =& g^{\lambda_\tau} w^{k_\tau} \\
=& g^{\tilde{\lambda}_\tau - \alpha_2 \langle M_\tau, \omega \rangle} \\
& \cdot \prod_{i \in [k]} (g^{a^2 c b_i})^{\frac{\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}} \cdot w^{\tilde{k}_\tau}, \\
K_{\tau,1} =& (z^{\rho(\tau)} h)^{-k_\tau} \\
=& (g^b)^{\langle M_\tau, \omega \rangle (\rho(\tau)\tilde{z} + \tilde{h})} \cdot (z^{\rho(\tau)} h)^{-\tilde{k}_\tau} \\
& \cdot \prod_{i \in [k]} (g^{acb_i})^{\frac{-(\rho(\tau)\tilde{z} + \tilde{h})\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}} \\
& \cdot \prod_{(i,j) \in [k,k]} (g^{(ac)^2 b_j / b_i})^{\frac{-\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_j^*)}} \\
& \cdot \prod_{i \in [k]} (g^{b^2 / b_i^2})^{\langle M_\tau, \omega \rangle (\rho(\tau) - A_i^*)} \\
& \cdot \prod_{(i,j) \in [k,k]} (g^{\frac{abcb_j}{b_i^2}})^{\frac{-\langle M_\tau, \omega \rangle (\rho(\tau) - A_i^*)}{(\rho(\tau) - A_j^*)}} \\
& \cdot \prod_{i \in [k]} (g^{\frac{abc\langle M_\tau, \omega \rangle}{b_i}}), \\
K_{\tau,2} =& g^{k_\tau} \\
=& (g^b)^{-\langle M_\tau, \omega \rangle} \cdot g^{\tilde{k}_\tau} \\
& \cdot \prod_{i \in [k]} (g^{acb_i})^{\frac{\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}}
\end{aligned}$$

by implicitly setting

$$k_\tau = -b\langle M_\tau, \omega \rangle + \sum_{i \in [k]} \frac{acb_i \langle M_\tau, \omega \rangle}{\rho(\tau) - A_i^*} + \tilde{k}_\tau.$$

Challenge: The attacker will output a pair of messages (m_0, m_1) of the same length. The challenger flips a random coin $\nu \leftarrow \{0, 1\}$ and implicitly sets $s = c$ from the q-type assumption. Also, set $r_\tau = b_\tau$ for every level $\tau \in [k]$. These parameters are properly distributed since $c, b_1, b_2, \cdots, b_q$ are information-theoretically hidden from the attacker's view. Now the challenger computes the following terms using the assumption:

$$\begin{aligned}
C =& m_\nu T,\ C_0 = g^s = g^c,\ C_{\tau,1} = g^{r_\tau} = g^{b_\tau}, \\
C_{\tau,2} =& (z^{A_\tau^*} h)^{r_\tau} w^{-s} \\
=& g^{b_\tau(\tilde{z} A_\tau^* + \tilde{h})} g^{-ac} \\
& \cdot \prod_{i \in [k]} g^{acb_\tau / b_i} \prod_{i \in [k]} g^{bb_\tau (A_\tau^* - A_i^*)/b_i^2} \\
=& (g^{b_\tau})^{\tilde{z} A_\tau^* + \tilde{h}} \prod_{i \in [k], i \neq \tau} g^{acb_\tau / b_i} \\
& \cdot \prod_{i \in [k], i \neq \tau} (g^{bb_\tau / b_i^2})^{A_\tau^* - A_i^*}.
\end{aligned}$$

For each $S_{i,j} \in CV_{R^*}$, set

$$\begin{aligned}
C_{i,j,1} &= v^{(L_i||d_j)s} = v^{(L_i||d_j)c}, \\
C_{i,j,2} &= (f^{(L_i||L_j)} v)^s = (f^{(L_i||L_j)} v)^c.
\end{aligned}$$

The challenger sends the ciphertext ct to $\mathcal{A}$, where

$$\begin{aligned}
ct =& (S^*, R^*, CV_{R^*}, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau \in [k]}, \\
& \{C_{i,j,t}\}_{S_{i,j} \in CV_{R^*}, t=1,2}).
\end{aligned}$$

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ outputs a guess ν' of ν. If $\nu' = \nu$ the challenger outputs $\mu' = 0$ and guesses the challenge term is $T = e(g, g)^{abc}$. Otherwise, the challenger outputs $\mu' = 1$ and guesses T is a random group element $\mathcal{R}$. It is obvious that the generation of public parameters and private keys is identical to the actual scheme.

In the case where $\mu = 1$ the attacker gains no information about ν. Therefore, we have $\Pr[\nu \neq \nu' | \mu = 1] = \frac{1}{2}$. Since the challenger guesses $\mu' = 1$ when $\nu \neq \nu'$, we have $\Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$ the attacker can see an encryption of M_ν. In this situation, the attacker's advantage is ε by definition. Thus we have $\Pr[\nu = \nu' | \mu = 0] = \frac{1}{2} + \varepsilon$. Since the challenger guesses $\mu' = 0$ when $\nu = \nu'$, we have $\Pr[\mu = \mu' | \mu = 0] = \frac{1}{2} + \varepsilon$.

Finally, the overall advantage of the challenger in the q-type game is

$$\frac{1}{2}\Pr[\mu = \mu' | \mu = 0] - \frac{1}{2} + \frac{1}{2}\Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}\varepsilon.$$

□

Theorem 2. *The proposed scheme is verifiable, if no polynomial time attacker $\mathcal{A}$ can get non-negligible advantage in the verifiability game defined in Section 4.3.*

Proof. We show that any polynomial time attacker $\mathcal{A}$ presents an incorrect updated ciphertext and succeeds in the verification with negligible probability.

The challenger proceeds the verifiability game, where $\mathcal{A}$ provides the attribute set S^*. The challenger runs $\mathbf{Setup}(1^\lambda)$ to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$. In the challenge phase, the challenger obtains a plaintext m and two revocation lists R and R^* from $\mathcal{A}$, where $R \subset R^*$. The challenger returns ct to the attacker by running $\mathbf{Encrypt}(m, S^*, R) \rightarrow ct$. $\mathcal{A}$ outputs the updated ciphertext ct^* to the challenger associated with the revocation list R^*.

Suppose that ct^* succeeds in the verification. That is, $\mathbf{Verify}(ct, ct^*) \rightarrow 1$. Let us consider the probability of $\mathcal{A}$ cheating with incorrect updated ciphertext.

Suppose the original ciphertext

$$\begin{aligned} ct =& (S^*, R, CV_R, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau\in[k]}, \\ & \{C_{i,j,t}\}_{S_{i,j}\in CV_R, t=1,2}). \end{aligned}$$

The updated ciphertext

$$\begin{aligned} ct^* =& (S^*, R^*, CV_{R^*}, C^*, C_0^*, \{C_{\tau,1}^*, C_{\tau,2}^*\}_{\tau\in[k]}, \\ & \{C_{i',j',t}^*\}_{S_{i',j'}\in CV_{R^*}, t=1,2}), \\ ct' =& (S^*, R^*, CV_{R^*}, C', C_0', \{C_{\tau,1}', C_{\tau,2}'\}_{\tau\in[k]}, \\ & \{C_{i',j',t}'\}_{S_{i',j'}\in CV_{R^*}, t=1,2}), \end{aligned}$$

where ct^* is the updated ciphertext returned by the attacker $\mathcal{A}$, when the revocation list R is changed to R^*, such that $R \subset R^*$, ct' is the updated ciphertext outputted by algorithm **update**.

Since $\mathbf{Verify}(ct, ct^*) \rightarrow 1$, then the following equations hold:

$$e(g, \frac{C_{\tau,2}^* C^*}{C}) e(C_0^*, w) = e(C_{\tau,1}^*, z^{A_\tau} h) e(\frac{C_0^*}{C_0}, e(g,g)^\alpha), \quad (1)$$

and

$$\begin{aligned} e(C_0^*, \prod_{k=1}^{\eta} (v^{(L_{i'_k} || d_{j'_k})})^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}), \quad (2) \\ e(C_0^*, \prod_{k=1}^{\eta} (f^{(L_{i'_k} || L_{j'_k})} v)^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,2}^*)^{c_k}), \end{aligned}$$

where $\tau \in [k], S_{i'_k,j'_k} \in CV_{R^*} - CV_R, 1 \leq k \leq \eta$ and $c_1, \cdots, c_\eta \leftarrow \mathbb{Z}_p$ are randomly selected by the challenger and unknown to $\mathcal{A}$.

Since $\mathbf{Verify}(ct, ct') \rightarrow 1$ (due to the correctness of the scheme), the following equations should hold:

$$e(g, \frac{C_{\tau,2}' C'}{C}) e(C_0', w) = e(C_{\tau,1}', z^{A_\tau} h) e(\frac{C_0'}{C_0}, e(g,g)^\alpha), \quad (3)$$

and

$$\begin{aligned} e(C_0', \prod_{k=1}^{\eta} (v^{(L_{i'_k} || d_{j'_k})})^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k}), \quad (4) \\ e(C_0', \prod_{k=1}^{\eta} (f^{(L_{i'_k} || L_{j'_k})} v)^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,2}')^{c_k}), \end{aligned}$$

where $\tau \in [k], \{S_{i'_k,j'_k}\}_{1\leq k\leq\eta}$ and $\{c_i\}_{1\leq i\leq\eta}$ are the same as the above.

According to Equations (1) and (3), we have $C' = C^*, C_0' = C_0^*, C_{\tau,1}' = C_{\tau,1}^*, C_{\tau,2}' = C_{\tau,2}^*$. In order to prove $ct' = ct^*$, we need to prove $\forall k, 1 \leq k \leq \eta, C_{i'_k,j'_k,1}' = C_{i'_k,j'_k,1}^*, C_{i'_k,j'_k,2}' = C_{i'_k,j'_k,2}^*$. We prove this by showing that the probability of $C_{i'_k,j'_k,1}' \neq C_{i'_k,j'_k,1}^*, C_{i'_k,j'_k,2}' \neq C_{i'_k,j'_k,2}^*$ for some k is negligible.

According to Equations (2) and (4), the following equations hold

$$\begin{aligned} e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}) \\ \Rightarrow \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k} &= \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}. \end{aligned}$$

Assume there exists a subset $L \subset [1, \eta]$ such that $\forall r \in L, C_{i'_r,j'_r,1}' \neq C_{i'_r,j'_r,1}^*$. Therefore, we remove the same items at both sides and get

$$\prod_{r\in L} (C_{i'_r,j'_r,1}')^{c_r} = \prod_{r\in L} (C_{i'_r,j'_r,1}^*)^{c_r}.$$

Set $C_{i'_r,j'_r,1}' = g^{\mu'_r}$ and $C_{i'_r,j'_r,1}^* = g^{\mu_r^*}$ for some unknown $\mu'_r, \mu_r^* \leftarrow \mathbb{Z}_p$ and $\mu'_r \neq \mu_r^*$, then we have

$$\prod_{r\in L} g^{c_r(\mu'_r - \mu_r^*)} = 1 \Rightarrow \sum_{r\in L} c_r(\mu'_r - \mu_r^*) = 0 (\text{mod } p).$$

Since c_r is unknown to $\mathcal{A}$, the probability of $C_{i'_r,j'_r,1}' \neq C_{i'_r,j'_r,1}^*$ is at most $1/p$, which is a negligible probability. Therefore, we have $\forall i, 1 \leq i \leq \eta, C_{i'_r,j'_r,1}' = C_{i'_r,j'_r,1}^*$. Similarly, $\forall i, 1 \leq i \leq \eta, C_{i'_r,j'_r,2}' = C_{i'_r,j'_r,2}^*$.

Hence, we have proved that $ct' = ct^*$ if $\mathbf{Verify}(ct, ct^*) \rightarrow 1$. That is, the attacker cannot generate an incorrect updated ciphertext while passing the verification. □

Table 2: Functionalities and features comparison of direct R-ABE

Schemes	Large universe	Ciphertext update	Security model	Verification
Attrapadung et al. [1]	×	–	standard model	×
Shi et al. [27]	×	anyone	random oracle model	√
Wang et al. [29] 1st scheme	×	–	standard model	×
Wang et al. [29] 2st scheme	√	–	standard model	×
Ours	√	anyone	standard model	√

Table 3: Efficiency comparison of direct R-ABE

Schemes	Size			Decryption cost	CTUpdate cost
	$\|pp\|$	$\|sk\|$	$\|ct\|$		
Attrapadung et al. [1]	$m+N+3$	$2(l+1)log(n)$	$\|S\|+1$ $+rlog(\frac{n}{r})$	$2(\|I\|+1)P$ $+(\|I\|+2)E$	–
Shi et al. [27]	$m+d+7$	$2l+2$	$2+\|S\|$ $+rlog(\frac{n}{r})$	$t_1P+\|I\|E$	t_2P
Wang et al. [29] 1st scheme	$2(m+2^d+1)$	$2l$	$2\|S\|+3$	$(l+2)P$ $+\|M\|E$	–
Wang et al. [29] 2st scheme	$2(m+2^d+1)$	$4l$	$2\|S\|+3$	$(2l+2)P$ $+\|M\|E$	–
Ours	6	$3l+2[log^2(n)$ $+log(n)]$	$2\|S\|+4r+1$	$(3\|I\|+4)P$ $+(\|I\|+1)E$	$4(r'+1)E$

7 Discussions

7.1 Performance Analysis

In this section, we compare the features and efficiency of our scheme with some existing direct R-ABE schemes [1, 27, 29]. This is shown in Table 2 and Table 3. Denote n to be the number of users in the system, m to be the maximum size of attributes, d to be the depth for all leaves in the full binary tree, l to be the size of rows in the LSSS matrix (or to be the size of the leaf nodes of access tree $\mathbb{A}$), I to be a subset of $\{1, 2, \cdots, l\}$, M to be a subset of the nodes in an access tree $\mathbb{A}$, r and r' to be the size of the revocation set R and R', respectively. N to be the maximum size of the cover set $cover(R)$. We denote

$$t_1 = 2|I| + d + 2 - depth(j),$$
$$t_2 = \sum_{j' \in cover(R')} (depth(j') - depth(j)).$$

E, P represent an exponentiation and pairing operation, respectively.

We first make a comparison in terms of functionalities and features in Table 2. Note that the second scheme presented by Wang et al. [29] and ours supporting large universe construction, but Wang et al.'s scheme [29] only achieved via fix a specific bound at stage of setup, and this approach places undesirable burden on the deployment of ABE schemes. Both Shi et al.'s scheme [27] and ours allow anyone to update ciphertexts and enable any auditor to verify whether the updated ciphertexts are correct. However, Shi et al.'s scheme [27] is secure in the random oracle model.

In addition, we make a analysis with respect to efficiency in Table 3. To the best of our knowledge, the storage overhead is mainly caused by storing the public parameters, the ciphertext and the secret key. It is easy to find that the sizes of the secret key and ciphertext in our scheme are close to the schemes [1, 27, 29]. However, the size of the public parameters is constant in our scheme. Meanwhile, due to the application of the SD method, our scheme could provide a smaller ciphertext component meant for performing revocation, at the cost of an admissible increased in the secret key size. On the other hand, although the efficiency of our scheme is lower than Attrapadung et al.'s scheme [1] and Wang et al.'s scheme [29] with reference to decryption, ours have more functionalities and features as analyzed above. In cihertext update, we use a trick similar to Shi et al.'s scheme [27], but our scheme is more efficiency: at most $4(r'+1)$ exponentiation operations. Whereas Shi et al.'s scheme [27] requires t_2 pairing operations.

7.2 Efficiency Improvement

As mentioned above, our scheme need to perform $3|I|+4$ bilinear pairing operations, which are considered the most expensive operation in pairing-based cryptographic protocols. These pairing operations lead to a heavy burden for the computation of decryption. Moreover, with the size of subset I increasing, the corresponding cost for the

computation of pairings would become higher. Therefore, it is indispensable to reduce the computation cost.

Outsourced ABE schemes might be the possible solution. Green et al. [11] first realized secure and efficient outsourcing of ABE decryption. Nowadays, many research works have been done on secure outsourced ABE schemes [13, 36]. Although outsourced ABE can reduce the computation cost of users, the systems have to introduce additional servers, which increases the system complexity [35].

Another possible solution is outsourcing computation techniques. Liu et al. [20] proposed a identity-based server-aided decryption scheme. Their scheme enables the receiver to decrypt the ciphertext without needing to compute paring with help of an external server. Based on the server-aided computation protocols, the notion of server-aided signature [7, 31] has been proposed in order to reduce the local computation. In 2014, Canard et al. [5] presented a secure and efficient delegating algorithm for pairing. This algorithm improved the computational complexity results. However, the solution was not feasible since exponentiation, membership test, and inversion were still required. Recently, Chen et al. [6] proposed an efficient and secure outsourcing algorithm for bilinear pairings in the two untrusted program model. In their scheme, the outsourcer never needs to perform any expensive operations such as exponentiations. It is not difficult to find that these secure outsourcing protocols and algorithms for bilinear pairings are also applicable to our scheme.

As a result, in our construction, we employ the secure outsourcing algorithm **Pair** presented by Chen et al. [6] to reduce the load of computation for the users. Similar to scheme [6], as a subroutine, the outsourcing algorithm **Pair** is invoked when users decrypt the ciphertexts. We show how the secure outsourcing algorithm **Pair** can be applied to our scheme. When a user decrypts a ciphertext ct, he runs the subroutine **Pair** to compute the message m (suppose the user identity $ID \notin R$ and the attribute set S satisfies the access structure (M, ρ)) as follows:

1) the user runs **Pair** to obtain $\mathbf{Pair}(C_0, k_{i,0}) \rightarrow \varphi_{i,0}$, $\mathbf{Pair}(C_{i,1}, k_{i,1}) \rightarrow \varphi_{i,1}, \mathbf{Pair}(C_{i,2}, k_{i,2}) \rightarrow \varphi_{i,2}$, $\mathbf{Pair}(C_0, D_{\tilde{i},\tilde{j},1}) \rightarrow \psi_1, \mathbf{Pair}(C_{i,j,1}, E_{\tilde{i},\tilde{j},1}) \rightarrow \psi_2$, $\mathbf{Pair}(C_0, D_{\tilde{i},\tilde{j},2}) \rightarrow \psi_3, \mathbf{Pair}(C_{i,j,2}, E_{\tilde{i},\tilde{j},2}) \rightarrow \psi_4$.

2) the user computes $B = \Pi_{i \in I}(\varphi_{i,0}\varphi_{i,1}\varphi_{i,2})^{\omega_i}$ and $B' = \psi_1\psi_2(\psi_3\psi_4)^{\frac{1}{(L_{\tilde{i}}||L_{\tilde{j}}) - (L_i||L_j)}}$.

3) the user computes $m = C/(BB')$ and outputs m.

Applying the secure outsourcing algorithm, the computation of decryption can be reduced to $|I| + 1$ exponentiation operations. This makes our system more suitable for practical applications.

8 Conclusions

In this paper, we have presented a directly revocable and verifiable key-policy attribute-based encryption (KP-ABE) scheme for large universe. Besides achieving efficient verification and large universe construction, the new scheme also utilizes the SD mechanism for direct revocation purpose. Compared with the CS scheme, the SD method greatly improves the broadcast efficiency and provides a smaller covering set, at the cost of an admissible increased in the secret key size. The DRV-KP-ABE scheme is selectively secure in the standard model.

Acknowledgments

We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Grants Nos. 61472470 and 61100229), and the Natural Science Basic Research Plan in Shaanxi Province of China (Grants Nos. 2014JM2-6091, 2015JQ1007 and 2015JQ6236).

References

[1] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Cryptography and Coding*, pp. 278–300, 2009.

[2] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography (Pairing'09)*, pp. 248–265, 2009.

[3] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Information Sciences*, vol. 276, pp. 354–362, 2014.

[4] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM conference on Computer and communications security*, pp. 417–426, 2008.

[5] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *Applied Cryptography and Network Security*, pp. 549–565, 2014.

[6] X. Chen, W. Susilo, J. Li, D. S. Wong, J. Ma, S. Tang, and Q. Tang, "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, vol. 562, pp. 112–121, 2015.

[7] S. S. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," *Information Security Technical Report*, vol. 17, no. 3, pp. 46–57, 2013.

[8] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[9] P. Datta, R. Dutta, and S. Mukhopadhyay, "Fully secure unbounded revocable attribute-based encryption in prime order bilinear groups via subset difference method," *IACR Cryptology ePrint Archive*, vol. 2015, pp. 293, 2015.

[10] X. Fu, S. Zeng, and F. Li, "Blind expressive ciphertext policy attribute based encryption for fine grained access control on the encrypted data," *International Journal Network Security*, vol. 17, no. 6, pp. 661–671, Nov. 2015.

[11] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of attribute-based encryption ciphertexts," in *Proceedings of the 20th USENIX Conference on Security (SEC'11)*, pp. 34, 2011.

[12] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography (PKC'14)*, pp. 293–310, 2014.

[13] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[14] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[15] K. Lee, "Self-updatable encryption with short public parameters and its extensions," *Designs, Codes and Cryptography*, vol. 79, pp. 121–161, 2016.

[16] K. Lee, D. H. Lee, and J. H. Park, "Efficient revocable identity-based encryption via subset difference methods," *IACR Cryptology ePrint Archive*, 2014.

[17] A. Lewko and B. Waters, "Unbounded hide identity-based encryption and attribute-based encryption," in *Advances in Cryptology (EUROCRYPT'11)*, pp. 547–567, 2011.

[18] K. Li and H. Ma, "Outsourcing decryption of multi-authority attribute-based encryption ciphertexts," *International Journal Network Security*, vol. 16, no. 4, pp. 286–294, 2014.

[19] Q. Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing key-policy attribute-based encryption scheme with constant-size ciphertext," *International Journal Network Security*, vol. 15, no. 3, pp. 161–170, 2013.

[20] J. K. Liu, C. K. Chu, and J. Zhou, "Identity-based server-aided decryption," in *Information Security and Privacy*, pp. 337–352, 2011.

[21] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal Network Security*, vol. 16, no. 6, pp. 437–443, 2014.

[22] Z. Liu and D. S. Wong, "Practical attribute based encryption: Traitor tracing, revocation, and large universe," *IACR Cryptology ePrint Archive*, 2014.

[23] Y. Ming, L. Fan, H. Jing-Li, and W. Zhao-Li, "An efficient attribute-based encryption scheme with revocation for outsourced data sharing control," in *First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 516–520, 2011.

[24] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM conference on Computer and Communications Security*, pp. 463–474, 2013.

[25] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology (CRYPTO'12)*, pp. 199–217, 2012.

[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, pp. 457–473, 2005.

[27] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Information Sciences*, vol. 295, pp. 221–231, 2015.

[28] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[29] P. Wang, D. Feng, and L. Zhang, "Towards attribute revocation in key-policy attribute based encryption," in *Cryptology and Network Security*, pp. 272–291, 2011.

[30] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[31] W. Wu, Y. Mu, W. Susilo, and X. Huang, "Provably secure server-aided verification signatures," *Computers and Mathematics with Applications*, vol. 61, no. 7, pp. 1705–1723, 2011.

[32] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

[33] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, "Attribute-based data sharing with flexible and direct revocation in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 11, 2014.

[34] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 511–516, 2013.

[35] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Efficient attribute-based data sharing in mobile clouds," *Pervasive and Mobile Computing*, 2014.

[36] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proceedings of IEEE INFOCOM*, pp. 522–530, 2014.

Hua Ma is a professor in the School of Mathematics and Statistics at Xidian University, Xian, China. Her research interests include design and analysis of fast public key cryptography, security theory and technology in electronic commerce, and network and information security.

Ting Peng is a master degree student in Mathematics at Xidian University. Her research focus on cryptography and network security.

Zhenhua Liu is an associate professor in the School of Mathematics and Statistics at Xidian University, Xian, China. His research interests include public key cryptography, cryptographic theory and security protocols in cloud computing.