Collusion Resistance CP-ABE Scheme with Accountability, Revocation and Privacy Preserving for Cloud-based E-health System

Zhenhua Liu[1,2], Yingying Ding[1,2], Ming Yuan[1,2], and Baocang Wang[3]
(Corresponding author: Yingying Ding)

School of Mathematics and Statistics, Xidian University[1]
Xi'an 710071, P.R. China
(Email: 2318326053@qq.com)
State Key Laboratory of Cryptology, P. O. Box 5159[2]
Beijing 100878, P.R. China
State Key Laboratory of Integrated Services Networks, Xidian University[3]
Xi'an 710071, P.R. China

(Received Oct. 6, 2021; Revised and Accepted Mar. 13, 2022; First Online Apr. 11, 2022)

Abstract

Cloud-based E-health systems can support users to store their health records in the cloud for better care, and ciphertext-policy attribute-based encryption (CP-ABE) with particular functionalities can enhance secure sharing. However, most traceable and revocable schemes only considered user traceability, and the revoked users could access data by conspiring with an unrevoked user. This paper presents a collision resistance CP-ABE scheme with accountability, revocation, and policy hiding. A user's decryption key is associated with the path in a binary tree and a self-selected secret value. By auditing the leaf node value and the secret value, a user or the authority is determined to take responsibility for a compromised key. Then using the binary tree can implement user revocation, which ensures collision avoidance and backward security. Furthermore, the security of the proposed scheme is proven, and the performance analysis indicates that the proposed scheme is efficient.

Keywords: Accountable Authority; Attribute-based Encryption; Collusion Resistance; User Revocation

1 Introduction

With the progress of cloud computing technology, cloud storage system has provided great convenience for users in data storage and sharing [28]. Hence, individuals and enterprises tend to outsource their data to the cloud for reducing storage costs. Due to the above characteristics, cloud storage system is appropriate for electronic health (E-health) system [24]. In order not to impede data sharing and ensure data security, a fine-grained access control system over encrypted data is urgently needed.

Attribute-based encryption is first described by Sahai and Waters [1], which can implement "one to many" access control. However, some malicious users existing in the system may reveal their decryption keys for some benefits. Since a decryption key is related to a user's attribute set, the user who had the same attribute set and divulged the decryption key cannot be identified. In order to settle the above problem, the concept of traceable CP-ABE [5] was proposed. Furthermore, a malicious user should be revoked immediately. In addition, most of the existing revocable CP-ABE schemes [12–14,25,26] supposed that the authority is fully trusted. However, the authority generates the decryption keys for all users and could also abuse the keys. Therefore, there are great expectations for a revocable CP-ABE with the authority accountability for the E-health system.

Moreover, considering that access policies in the form of plaintext are coupled with ciphertext and stored directly in the cloud, it is inevitable to reveal the sensitive information of patients in the E-health system [4]. In order to enhance the privacy protection of users in the E-health system, the CP-ABE schemes that can implement hidden policy were proposed [11,17].

1.1 Related Works

In order to track down malicious users, Li *et al.* presented the first accountable CP-ABE scheme [6] that supports AND-gate policies. To enhance the expressiveness of access policies, Liu *et al.* constructed a white-box traceable CP-ABE scheme supporting any monotone access structures [5] and a black-box traceable CP-ABE scheme [7] in 2013. However, Liu *et al.*'s two accountable schemes were structured by utilizing bilinear groups of composite order that were inefficient. Later, a white-box traceable CP-ABE schemes were proposed by Ning *et al.* [8], which based on bilinear groups of prime order.Unfortunately, the above schemes only offered the solution of user key abuse. Nevertheless, the authority can generate decryption keys for illegal users without the threats of being caught. Out of that reason, Ning *et al.* designed the first accountable authority CP-ABE scheme [9] based on the bilinear groups of composite order, which can support flexible access policies. Later, Zhang *et al.* described a CP-ABE scheme [10] based on LSSS that supports the authority accountability, and Li *et al.* presented an accountable authority CP-ABE scheme [11] with hidden policy by utilizing the bilinear groups of prime order. But none of the above schemes took into account user revocation. Considering, the privilege for the baleful user to decrypt a ciphertext should be revoked immediately. Thus, a secure revocation CP-ABE scheme needs to be proposed to revoke the malicious user.

To revoke the illegal user, Hur *et al.* constructed an indirect user revocation CP-ABE scheme [13] based on a more expressive access tree structure, where a secret key or decryption key includes two parts. Unfortunately, Hur *et al.*'s scheme suffered from collusion attacks. In order to avoid user collusion, Li *et al.* [14] proposed a CP-ABE scheme, in which group secret key and private key are bound together by embedding the same random values in two keys. To improve the expressiveness, Lee *et al.* put forward a revocable CP-ABE scheme [15] that can support LSSS access policy. Recently, Ning *et al.* [16] structured two schemes with authority accountability and user revocation: ATER-CP-ABE and ATIR-CP-ABE, where the former realizes revocation by the revocation list and the later implements revocation by key update, and Han *et al.* proposed a scheme [17] with traceability and user revocation. However, their schemes cannot implement the backward security, which means that the previous ciphertext cannot be decrypted by the revoked user [4]. At the same time, the above schemes fail to consider such a problem that the access structure stored directly in the cloud could reveal user sensitive information in the E-health system.

Although numerous of ABE schemes have been presented to protect users' privacy data, sensitive information carried by access policies in ciphertext will still expose users' privacy. To prevent the above problem, many hidden-policy CP-ABE schemes [18,19] were constructed.

1.2 Our Motivation and Contributions

Han *et al.* [17] proposed a multi-purpose CP-ABE scheme, which implemented user accountability, user revocation and hidden policy. However, their scheme could encounter with the collusion attacks that the revoked users can cooperate with the unrevoked users to decrypt the ciphertext and required the authority is full trusted. Though the schemes of Ning *et al.* [16] and Li *et al.* [11] considered the semi-trusted authority, the former failed to guarantee the backward security, while the later cannot adopt the flexible LSSS access policies and achieve user revocability.

1.2.1 Comments on Han *et al.*'s Scheme

In Han *et al.*'s scheme [17], suppose that there are two users Alice and Bob, where the former has the decryption key ($K' = c$, $K = g^{\frac{\alpha}{a+c}} \cdot h^r$, $L = g^r$, $L' = g^{a \cdot r}$, $\{K_\tau = g^{s_\tau \cdot r} \cdot u^{-(a+c) \cdot r}\}_{\tau \in I_S}$, $K_A = g^{\frac{r}{x_{i_d}}}$, $\{x_i\}_{i \in path(i_d)}, \mathcal{S}$) and the latter has the decryption key ($(K')^* = c^*$, $K^* = g^{\frac{\alpha}{a+c^*}} \cdot h^{r^*}$, $L^* = g^{r^*}$, $(L')^* = g^{a \cdot r^*}$, $\{K_\tau^* = g^{s_\tau \cdot r^*} \cdot u^{-(a+c^*) \cdot r^*}\}_{\tau \in I_S^*}$, $K_B = g^{\frac{r^*}{x_{i_d^*}}}$, $\{x_i\}_{i \in path(i_d^*)}, \mathcal{S}^*$). They want to access the ciphertext ($C = m \cdot e(g,g)^{\alpha s}$, $C_0 = g^s$, $C_0' = g^{a \cdot s}$, $\{C_{i,1} = h^{\lambda_i} \cdot u^{k_i}$, $C_{i,2} = g^{-k_i \cdot t_{\rho(i)} + \lambda_i}$, $C_{i,3} = g^{k_i}\}_{i \in [1,l]}$, $\{T_j = y_j^s\}_{j \in cover(\mathcal{R})}, \mathcal{R}, \overline{W}$), where the set $cover(\mathcal{R})$ means the minimum cover set associated with the revocation list $\mathcal{R}$. Alice is an unrevoked user, but her attribute set $\mathcal{S}$ does not meet $\overline{W}$. On the other hand, Bob's attribute set $\mathcal{S}^*$ meets $\overline{W}$, but as a revoked user, Bob cannot also decrypt the ciphertext since there are no elements in $cover(\mathcal{R}) \cap path(i_d^*)$. Furthermore, none of $\{x_i\}_{i \in path(i_d^*)}$ can be used to decrypt the ciphertext. Although they fail to decrypt the ciphertext individually, they can successfully decrypt the ciphertext if they combine their respective decryption keys. Since Alice is not be revoked, she can obtain the node $j \in cover(\mathcal{R}) \cap path(i_d)$ and give x_j to Bob. Once Bob obtains x_j, and since his attribute set $\mathcal{S}^*$ meets $\overline{W}$, he can decrypt the ciphertext as follows:

1) Firstly, use x_j to compute $\frac{x_{i_d^*}}{x_j}$, then compute $B = e(K_B, T_j)^{\frac{x_{i_d^*}}{x_j}} = e(g,g)^{r^* \cdot s}$.

2) Secondly, calculate E, F, and D:

$$\begin{aligned} E =& [e((L^*)^{(K')^*} \cdot (L')^*, C_{i,1}) \cdot e(L^*, C_{i,2}) \\ & \cdot e(K^*_{\rho(i)}, C_{i,3})] \\ =& e(g,h)^{(a+c^*) \cdot r^* \cdot \lambda_i} \cdot e(g,g)^{r^* \cdot \lambda_i}, \\ F =& \prod_{i \in I} (E)^{c_i} = e(g,h)^{(a+c^*) \cdot r^* \cdot s} \cdot e(g,g)^{r^* \cdot s}, \\ D =& e(K^*, C_0^{(K')^*} \cdot C_0') \\ =& e(g,g)^{\alpha \cdot s} \cdot e(g,h)^{(a+c^*) \cdot r^* \cdot s}. \end{aligned}$$

3) Finally, recover the message $m = \frac{C \cdot F}{D \cdot B}$.

Moreover, Bob can decrypt the ciphertext without anyone's help, just by changing the decryption algorithm a little. B is computed as follows:

$$B = e(K_B, C_0)^{x_{i_d^*}} = e(g,g)^{r^* \cdot s},$$

and E, F, D are calculated by the same way as before. Thus Bob can successfully get the plaintext. Due to the aforementioned flaws, Han *et al.*'s scheme cannot support the backward security and forward security, where the forward security means the revoked user cannot decrypt the ciphertext in the future [4].

1.2.2 Our Contributions

Inspired with Han *et al.*'s scheme [17] and Li *et al.*'s scheme [11], an accountable and revocable CP-ABE (AR-CP-ABE) scheme with privacy protection is proposed, which can provide the authority and user accountability, the direct user revocation, the backward security, and the partial hidden policy. The main contributions and techniques are as follows:

1) **Authority and user accountability.** We construct a CP-ABE scheme with the authority and the user accountability based on the bilinear groups of prime order and flexible LSSS access policies. In the proposed scheme, the full decryption key of a user contains the partial keys generated by the authority based on the attribute set and the path in a binary tree that one leaf node value corresponds to one user, which is used to trace the user, and a secret value chosen by the user, which is used to ultimately determine whether the compromised key was generated by the user or the authority.

2) **User collusion avoidance.** In the proposed scheme, we bind the values of the node on the user path to the value in connection with the user's identity, which can avoid the collusion between the unrevoked user and the revoked user. Furthermore, we define the collusion resistance security model between the revoked user and the unrevoked user, and give a detailed proof that our scheme can resist the user collusion attack.

3) **Backward security and forward security.** In Han *et al.*'s scheme [17], the revoked user can obtain the plaintext with knowing the value x_{i_d} of node, which could break the backward security and forward security. To solve this problem, we embed x_{i_d} into the key instead of sending $\{x_i\}_{i \in path(i_d)}$ directly to the user, and then update the previous or old ciphertext.

1.3 Organization

The remainder of the paper is structured as follows. In Section 2, we first recall some background knowledge used in this paper. In Section 3, the formal definition and security model of AR-CP-ABE are given. The detailed construction of the proposed scheme is described in Section 4 and the security proof in Section 5. In Section 6, we compare our work with the other related works on functionalities and efficiency. Finally, in Section 7, a conclusion and the future work are given.

2 Preliminaries

2.1 Linear Secret Sharing Scheme

Suppose that $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_n\}$ is an attribute name universe, and for $\forall \mathcal{L}_i \in \mathcal{L}$, the set of attribute value is $\mathcal{L}_i = \{a_{i,1}, a_{i,2}, \cdots a_{i,n_i}\}$. A linear secret-sharing scheme (LSSS) [17] can stand for an access control policy by (M, ρ), where M is an $l \times n$ matrix and ρ is a mapping from the rows of M to attribute names in $\mathcal{L}$. The LSSS comprises of two algorithm:

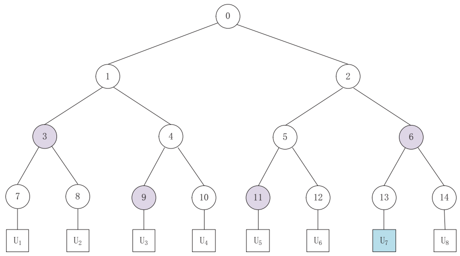


Figure 1: Binary tree $\mathcal{T}$

1) **Share** s**.** The purpose of the algorithm is to hide a value $s \in \mathbb{Z}_p$. Choose a vector $\vec{v} = (s, v_2, \cdots, v_n)^\top$, where $v_2, \cdots, v_n \in \mathbb{Z}_p$ are chosen randomly. Compute $\lambda_i = M_i \cdot \vec{v}$ as a sharing of s, where λ_i matches with the attribute name $\rho(i)$, and M_i is the i-th row vector of M.

2) **Reconstruct** s**.** The algorithm is utilized to recover s. Suppose that S is an authorized set, where S satisfies the access policy (M, ρ) and $I = \{i : \rho(i) \in I_S\} \subseteq \{1, 2, \cdots, l\}$. Then, some coefficients $\{c_i | i \in I\}$ such that $\sum\limits_{i \in I} c_i \lambda_i = s$ will be found.

Set $\mathcal{S} = (I_S, S)$ as a user attribute set, where $I_S \subseteq \mathcal{L}$ is a set of user attribute name, and $S = \{s_i\}_{i \in I_S}$ stands for attribute values. Furthermore, the access policy is represented by $W = (M, \rho, T)$, where M is an $l \times n$ matrix, ρ is a mapping from rows of M to attribute names in $\mathcal{L}$ that each attribute name can occur only once, and $T = \{t_{\rho(i)}\}_{i \in [1,l]}$ is the attribute value related to (M, ρ). Let $\mathcal{S} \in W$ denote $\mathcal{S}$ matches W, which means that there exists a set $I = \{i : \rho(i) \in I_S\} \subseteq \{1, 2, \cdots, l\}$ satisfying W and for $\forall i \in I$, $s_{\rho(i)} = t_{\rho(i)}$, and $\mathcal{S} \notin W$ denote $\mathcal{S}$ dose not match W. Finally, the access policy removing the attribute values is represented by $\overline{W} = (M, \rho)$.

2.2 Binary Tree

A set of all users in the system and a revocation list are represented by $\mathcal{U}$ and $\mathcal{R}$, respectively. A binary tree $\mathcal{T}$ [20] is described as:

- Each leaf node is related to a user U. Set $|\mathcal{U}|$ as the total number of users. Supposed that the nodes are numbered by natural numbers. Concretely, 0 is the serial number of the root node and $2|\mathcal{U}| - 2$ is the last.

- Let *path*(i) be a path from the root node to the node related to i and *cover*$(\mathcal{R})$ be a minimum set of the nodes that can cover all users that are not in $\mathcal{R}$, we call it the minimum cover set.

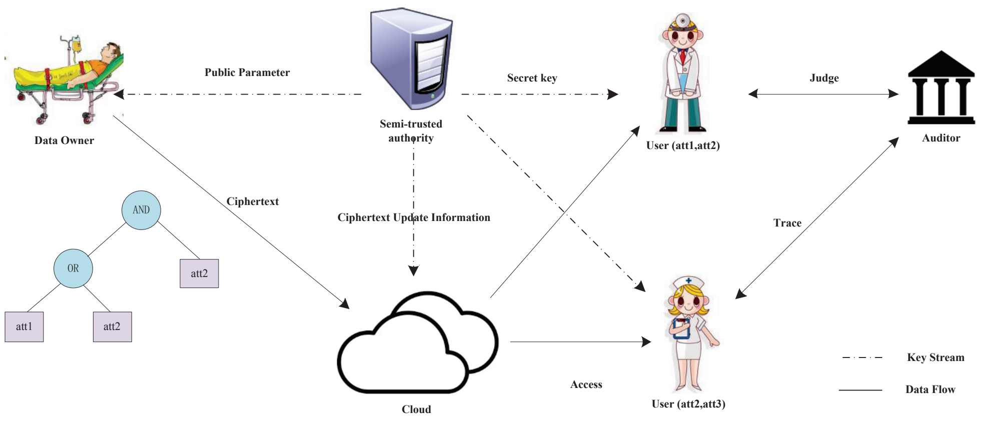


Figure 2: System construction of AR-CP-ABE

As depicted in Figure 1, given a revocation list $\mathcal{R} = \{U_4, U_6\} = \{10, 12\}$, then $cover(\mathcal{R}) = \{3, 9, 11, 6\}$. From the tree, the path of $U_7 : path(U_7) = path(13) = \{0, 2, 6, 13\}$ can be obtained. Thus, the intersection $j = cover(\mathcal{R}) \cap path(U_7) = \{6\}$ can be computed.

2.3 Hardness Assumptions

Now we review three well-known complexity assumptions [21–23] that the security of the proposed scheme are reducible to.

Definition 1. *Let $\mathbb{G}$ be a multiplication cyclic groups of prime order p, and g be a generator of $\mathbb{G}$. Given a tuple (g, g^z), where $z \in \mathbb{Z}_p^*$, the Discrete Logarithm Problem (DLP) is to output z. Furthermore, the DLP hardness assumption holds if no PPT adversary $\mathcal{A}$ can calculate z with non-negligible advantage.*

Definition 2. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplication cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Given $Y = (g, g^s, g^d, g^{d^2}, \cdots, g^{d^q}, g^{d^{q+2}}, \cdots, g^{d^{2q}})$, where $s, d \in \mathbb{Z}_p^*$, the q-Bilinear Diffie-Hellman Exponent (q-BDHE) problem is to distinguish $e(g, g)^{d^{q+1} \cdot s}$ from an element Z that is selected in $\mathbb{G}_T$ randomly. Moreover, the q-BDHE hardness assumption holds if no PPT adversary $\mathcal{A}$ can solve the q-BDHE problem with non-negligible advantage.*

Definition 3. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplication cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Given $(l+1)$-tuple $(g, g^x, g^{x^2}, \cdots, g^{x^l})$, where $x \in \mathbb{Z}_p^*$, the l-Strong Diffie-Hellman (l-SDH) problem is to output a tuple $(c, g^{1/(a+c)})$. Furthermore, the l-SDH hardness assumption holds if no PPT adversary $\mathcal{A}$ can calculate $(c, g^{1/(a+c)})$ with non-negligible advantage.*

3 System and Security Models

In this section, the system architecture, the formal definition, and a series of security models about AR-CP-ABE will be given.

3.1 System Framework

There are five entities in the system framework of AR-CP-ABE, as depicted in Figure 2.

- **Semi-trusted authority**. A semi-trusted authority can setup the system, publish the public parameters, and generate the decryption keys and the update keys for the users and cloud server, respectively.
- **Data owner (Patient)**. The data owners can encrypt the health record according to the specified access policy to the cloud.
- **Cloud server**. A cloud, which is honest-but-curious, can store the ciphertext for the data owner and update the ciphertext by using of the update key from the authority.
- **User (Medical personnel and specialists, etc)**. A user can decrypt the ciphertext successfully when the user's attributes can satisfy the access policy and identity is not in the revocation list.
- **Auditor**. A trusted auditor is responsible for the audit and revocation procedure, and returns the corresponding results to the users.

3.2 The Formal Definition of AR-CP-ABE

A formal AR-CP-ABE scheme mainly includes seven algorithms as follows:

- **Setup**$(\lambda, \mathcal{T}, \mathcal{L}) \to (PP, MSK)$. The algorithm is executed by the authority. Take as input the security parameter λ, a binary tree $\mathcal{T}$ and the attribute universe $\mathcal{L}$, then output the public parameters PP and the master secret key MSK that is kept secretly.

- **KeyGen**$(MSK, U, \mathcal{S}) \rightarrow SK$. The authority interacts with a user U, and runs the algorithm. The authority inputs the master secret key MSK, the user U and its attribute set $\mathcal{S} = (I_S, S)$, then sends the intermediate key for the user.
- **Encryption**$(PP, m, (M, \rho, T), \mathcal{R}) \rightarrow CT$. The algorithm is executed by the data owner that inputs the public parameter PP, a message m, the newly revocation list $\mathcal{R}$, and the access policy $W = (M, \rho, T)$, and generates a ciphertext CT.
- **Decryption**$(SK, CT) \rightarrow m$. After inputting the ciphertxt CT and her or his decryption key SK, the ciphertext can be decrypted successfully when the user's attributes can satisfy the access policy and identity is not in $\mathcal{R}$.
- **KeySanityCheck**$(SK) \rightarrow 1/\perp$. As a third party, the auditor runs this algorithm to evaluate whether the key is formal well.
- **Trace**$(SK_{suspected}, PP, \mathcal{R}) \rightarrow U/authority/\perp$. The auditor executes this algorithm and outputs who is a dishonest party, and manages the revocation list $\mathcal{R}$.
- **CTUpdate**$(CT, R', X') \rightarrow CT'$. The ciphertext update algorithm is executed by the cloud. Take the ciphertext CT, the new revocation list $\mathcal{R}'$ and the update key X' from the authority as input, and output an updated ciphertext CT'.

3.3 IND-CPA Security Model

The IND-CPA security [17] of AR-CP-ABE scheme is depicted by a game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. Specific steps are as follow:

- **Initialization**: $\mathcal{A}$ determines a challenged access policy $W^* = (M^*, \rho^*, T^*)$ and a revocation list $\mathcal{R}^*$, where M^* is an $l^* \times n^*$ matrix with $n^* \leq q$, ρ is a mapping from rows of M to attribute names in $\mathcal{L}$, and $T^* = \{t_{\rho^*(i)}\}_{i \in [1, l^*]}$ is the attribute value related to (M^*, ρ^*).
- **Setup**: The challenger $\mathcal{C}$ generates a master secret key MSK and public parameters PP, and submits PP to $\mathcal{A}$ by utilizing the Setup algorithm.
- **Phase 1**: In this phase, the adversary $\mathcal{A}$ submits a series of user attribute sets $\{(U_1, \mathcal{S}_1), \cdots, (U_q, \mathcal{S}_q)\}$ to $\mathcal{C}$.
  - **Case 1**: If $\mathcal{S}_i \in W^*$ and $U \notin \mathcal{R}^*$, then $\mathcal{C}$ aborts.
  - **Case 2**: If $\mathcal{S}_i \notin W^*$ or $U \in \mathcal{R}^*$, $\mathcal{C}$ generates the intermediate keys for $\mathcal{A}$ by running the KeyGen algorithm.
- **Challenge**: $\mathcal{A}$ chooses two equal-length messages m_0, m_1 and sends them to $\mathcal{C}$. Then $\mathcal{C}$ flips a coin $\upsilon \in \{0, 1\}$ randomly and encrypts m_υ under the access policy (M^*, ρ^*) and the revocation list $\mathcal{R}^*$. Finally, the ciphertext CT^* will be sent to $\mathcal{A}$ by $\mathcal{C}$.
- **Phase 2**: Phase 2 is as same as Phase 1.
- **Guess**: $\mathcal{A}$ outputs a guess υ' of υ. $\mathcal{A}$ will win the game if $\upsilon' = \upsilon$.

The advantage of $\mathcal{A}$ that wins the above game is defined as

$$Adv(\mathcal{A}) = |\Pr[\upsilon' = \upsilon] - 1/2|.$$

Definition 4. *The AR-CP-ABE scheme is IND-CPA secure if all the PPT adversaries have at most negligible advantage in the above game.*

3.4 Accountability Security Model

The accountability security model that used by Ning *et al.* [11] includes three security games: *Dishonest-Authority* game, *Dishonest-User-I* game and *Dishonest-User-II* game. We also use the accountability security model in the proposed scheme.

1) ***Dishonest-Authority game***. The meaning of the game is that the adversarial authority attempts to forge user's key family number ω in the user's decryption key. The *Dishonest-Authority* game for the proposed scheme proceeds as follows.

   - **Setup**: The adversary $\mathcal{A}$ (dishonest authority) submits the public parameters PP to $\mathcal{C}$ by calling the Setup algorithm.
   - **Key Generation**: $\mathcal{A}$ invokes the KeyGen algorithm to generate a intermediate key for $\mathcal{C}$. $\mathcal{C}$ can abort the game when intermediate key is not well-formed.
   - **Key Forgery**: $\mathcal{A}$ outputs a forged decryption key SK' associated with U. Then $\mathcal{C}$ checks whether SK' is well-formed. The $\mathcal{C}$ can abort the game if SK' is not well-formed.

   Suppose that the event that the adversary wins the game is represented by ζ. The advantage of the adversary in *Dishonest-Authority* game is defined as

   $$Adv(\mathcal{A}) = \Pr[\zeta].$$

2) ***Dishonest-User-I game***. The intuition under the game is that a decryption key of a new user U cannot be forged by an adversarial user. The *Dishonest-User-I* game will be carried out as follows.

   - **Setup:** The challenger $\mathcal{C}$ generates a master key MSK and submits the public parameters PP by calling the Setup algorithm.
   - **Key Query:** $\mathcal{A}$ submits the attribute sets $\{(U_1, \mathcal{S}_1), \cdots, (U_q, \mathcal{S}_q)\}$ to $\mathcal{C}$ for requesting the intermediate keys. Then $\mathcal{C}$ invokes the KeyGen algorithm to generate the intermediate keys.
   - **Key Forgery:** $\mathcal{A}$ outputs a forged key SK^*. If $Trace(SK^*, \mathcal{R}, PP) \neq \perp$ and $Trace(SK^*, \mathcal{R}, PP) \notin \{U_1, \cdots, U_q\}$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ in the above game is defined as

$$Adv(\mathcal{A}) = \Pr[Trace(SK^*, \mathcal{R}, PP) \neq \perp \cup Trace(SK^*, \mathcal{R}, PP) \notin \{U_1, \cdots, U_q\}].$$

3) ***Dishonest-User-II game***. The meaning of the game is that another key family number (represented by ω) cannot be forged by an adversarial user. The *Dishonest-User-II* game for the proposed scheme will be carried out as follows.

- **Setup:** The challenger $\mathcal{C}$ generates a master secret key MSK and the public parameters PP by executing the Setup algorithm. Then $\mathcal{C}$ submits PP to $\mathcal{A}$.
- **Key Query:** $\mathcal{A}$ submits attribute sets $\{(U_1, \mathcal{S}_1), \cdots, (U_q, \mathcal{S}_q)\}$ to $\mathcal{C}$. Then $\mathcal{C}$ generates the intermediate keys for $\mathcal{A}$ by calling the KeyGen algorithm.
- **Key Forgery:** $\mathcal{A}$ outputs a forged key SK^* of the user U. $\mathcal{A}$ wins the game if $(U, c) = (U_i, c_i) \in \{(U_1, c_1), \cdots, (U_q, c_q)\}, \omega \neq \omega_i$ and SK is well-formed.

The advantage of $\mathcal{A}$ in the above game is defined as

$$Adv(\mathcal{A}) = \Pr[Trace(SK^*, \mathcal{R}, PP) \in \{U_1, \cdots, U_q\} \cup Audit(SK^*) \rightarrow innocent].$$

Definition 5. *The AR-CP-ABE scheme is accountable if all the PPT adversaries have at most negligible advantage in the above three games.*

3.5 Collusion Resistance Security Model

In order to apply to the proposed scheme with user revocation, we modify the security model of Li *et al.* [14] by replacing the attribute with the user. The game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is defined as follows:

- **Initialization**: $\mathcal{A}$ sends a challenged access policy $W^* = (M^*, \rho^*, T^*)$ and a revocation list $\mathcal{R}^*$ to $\mathcal{C}$.
- **Setup**: The challenger $\mathcal{C}$ generates a master secret key MSK and public parameters PP, and submits PP to $\mathcal{A}$ by calling the Setup algorithm.
- **Phase 1**: In this phase, the adversary $\mathcal{A}$ can issue two types of queries as follows:
  - **Type-I key query** $\langle U_I, \mathcal{S}_I \rangle$: User attribute set $\mathcal{S}_I \notin W^*$, but the user U_I is unrevoked. $\mathcal{C}$ interacts with $\mathcal{A}$ and then generates a decryption key by running the KeyGen algorithm. Finally, $\mathcal{C}$ returns the key to $\mathcal{A}$.
  - **Type-II key query** $\langle U_{II}, \mathcal{S}_{II} \rangle$: User U_{II} already has been revoked, while the user attribute set $\mathcal{S}_{II} \in W^*$. $\mathcal{C}$ interacts with $\mathcal{A}$ and then generates a decryption key by running the KeyGen algorithm. Later, $\mathcal{C}$ sends the key to $\mathcal{A}$.
- **Challenge**: After receiving two equal-length messages m_0, m_1 from $\mathcal{A}$, $\mathcal{C}$ flips a coin $b \in \{0, 1\}$ randomly and encrypts m_b by using the challenged access policy W^* and the revocation list $\mathcal{R}^*$. Finally, the ciphertext CT^* will be sent to $\mathcal{A}$.
- **Phase 2**: Phase 2 is as same as Phase 1.
- **Guess**: $\mathcal{A}$ outputs a guess b' of b. $\mathcal{A}$ will win the game if $b' = b$.

The advantage of $\mathcal{A}$ that wins the above game is defined as

$$Adv(\mathcal{A}) = |\Pr[b' = b] - 1/2|.$$

Definition 6. *The AR-CP-ABE scheme with user revocation is secure against collusion attacks if all the PPT adversaries have at most negligible advantage in the above game.*

4 Construction of AR-CP-ABE

Inspired with Han *et al.*'s scheme [17] and Li *et al.*'s scheme [11], we will construct an AR-CP-ABE scheme in this section.

- **Setup**$(\lambda, \mathcal{L}, \mathcal{T}) \rightarrow (PP, MSK)$. The algorithm is executed by the authority, and takes as input a security parameter λ, an attribute universe $\mathcal{L}$ and a binary tree $\mathcal{T}$ associated with user U that $U \in \mathcal{U}$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplication cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear mapping. The algorithm is implemented as follows:

  1) Pick $h, u \in \mathbb{G}$ and $a, \alpha \in \mathbb{Z}_p$ randomly.
  2) For each node of $\mathcal{T}$, randomly select $\{x_i\}_{i=0}^{2|\mathcal{U}|-2} \in \mathbb{Z}_p^*$, and compute $\{y_i = g^{x_i}\}_{i=0}^{2|\mathcal{U}|-2}$.
  3) Select a probabilistic symmetric encryption scheme (Enc, Dec) from $\{0,1\}^*$ to $\mathbb{Z}_p$, which sets $\bar{k} \in \mathbb{Z}_p$ as secret key. Then, the authority sends $\bar{k}$ to the auditor.

  Finally, the public parameters are published as:

  $$PP = (p, \mathbb{G}, \mathbb{G}_T, e, g, h, u, e(g,g)^{\alpha}, g^a, \{y_i\}_{i=0}^{2|\mathcal{U}|-2}),$$

  and the master key is kept secretly as:

  $$MSK = (a, \alpha, \{x_i\}_{i=0}^{2|\mathcal{U}|-2}, \bar{k}).$$

- **KeyGen**$(MSK, U, \mathcal{S}) \rightarrow (SK)$. The authority interacts with a user U whose attribute set is $\mathcal{S} = (I_S, S)$, where I_S is a set of user attribute name, and $S = \{s_i\}_{i \in I_S}$ stands for a set of attribute values. Then, the authority computes $c = Enc_{\bar{k}}(i_d)$, where i_d is the value of the leaf node about the user U. The algorithm is executed as follows:

1) The user randomly chooses $\omega \in \mathbb{Z}_p^*$, computes $H = h^{\omega}$, and sends H to the authority. The user also needs to make a proof of knowledge to the authority with regard to the discrete logarithm of H.
2) If the proof of knowledge is valid, the authority selects $r \in \mathbb{Z}_p$, and for $\forall \tau \in I_S$, computes as follows: $(K' = c, K = g^{\frac{\alpha}{a+c}} \cdot H^r, L = g^r, L' = g^{a \cdot r}, L'' = g^{r \cdot x_{i_d}}, \{K_\tau = g^{x_{i_d} \cdot s_\tau \cdot r} \cdot u^{-(a+c) \cdot r}\}_{\tau \in I_S})$.
3) Suppose $path(i_d) = \{i_0, \cdots, i_d\}$, where $i_0 = root$ and i_d is the value of a leaf node about the user U in the tree. The authority computes $\{KU_i = g^{r \cdot \frac{x_{i_d}}{x_i}}\}_{i \in path(i_d)}$ for the user U, then sends a tuple (U, i_d) and the intermediate key $(K', K, L, L', L'', \{KU_i\}_{i \in path(i_d)}, \{K_\tau\}_{\tau \in I_S}, \mathcal{S})$ to the auditor and the user, respectively.
4) Finally, the user sets the full decryption key $SK = (K', K, T' = \omega, L, L', L'', \{K_\tau\}_{\tau \in I_S}, \{KU_i\}_{i \in path(i_d)}, \mathcal{S})$, and the auditor adds the tuple (U, i_d) in the list LN that is used to trace.

- **Encryption**$(PP, m, (M, \rho, T), \mathcal{R}) \rightarrow CT$. Taking as input the public parameters PP, a message $m \in \mathbb{G}_T$, the latest revocation list $\mathcal{R}$, and an access policy $W = (M, \rho, T)$, where $T = \{t_{\rho(i)}\}_{i \in [1,l]}$ is the attribute value, a data owner runs the algorithm in the following.

1) Choose a vector $\vec{v} = (s, v_2, \cdots, v_n)^\top$ randomly, where $s, v_2, \cdots, v_n \in \mathbb{Z}_p$, and calculate $\vec{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_l)^\top = M\vec{v}$.
2) Select $k_i \in \mathbb{Z}_p$ randomly, where $i \in [1, l]$, and compute a partial ciphertext based on the access policy W: $(C = m \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{a \cdot s}, \{C_{i,1} = h^{\lambda_i} \cdot u^{k_i}, C_{i,2} = g^{-k_i \cdot t_{\rho(i)} + \lambda_i}, C_{i,3} = g^{k_i}\}_{i \in [1,l]})$.
3) Compute a partial ciphertext related to the revocation list $\mathcal{R}$: $(\{T_j = y_j^s\}_{j \in cover(\mathcal{R})})$.
4) Finally, send a full ciphertext as follows: $CT = (C, C_0, C_0', \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [1,l]}, \{T_j\}_{j \in cover(\mathcal{R})}, \mathcal{R}, \overline{W})$, where $\overline{W} = (M, \rho)$ is the access policy that removes the attribute value set.

- **Decryption** $(SK, CT) \rightarrow m$. Taking the ciphertxt CT as input, the user who owns the full decryption key $SK = (\mathcal{S}, K', K, T', L, L', L'', \{KU_i\}_{i \in path(i_d)}, \{K_\tau\}_{\tau \in I_S})$ and the attribute set $\mathcal{S}$ can implement the following algorithm.

1) For $U \notin \mathcal{R}$, there must exist a node j that $j \in cover(\mathcal{R}) \cap path(U)$. Suppose that $path(U) = \{i_0, \cdots, i_{dep(j)}, \cdots, i_d\}$, where $i_{dep(j)} = j$, and compute
$$B = e(KU_j, T_j)^{T'} = e(g,g)^{r \cdot x_{i_d} \cdot s \cdot \omega}.$$
2) Let $I = \{i : \rho(i) \in I_S\} \subseteq [1, 2, \cdots, l]$. There exist coefficients $\{c_i | i \in I\}$ such that $\sum_{i \in I} c_i \lambda_i = s$. And then compute
$$\begin{aligned} E =& [e(L^{K'} \cdot L', C_{i,1}) \cdot e(L'', C_{i,2}) \cdot e(K_{\rho(i)}, C_{i,3})]^{T'} \\ =& e(g,h)^{(a+c) \cdot r \cdot \lambda_i \cdot \omega} \cdot e(g,g)^{r \cdot x_{i_d} \cdot \lambda_i \cdot \omega}, \\ F =& \prod_{i \in I} (E)^{c_i} = e(g,h)^{(a+c) \cdot r \cdot s \cdot \omega} \cdot e(g,g)^{r \cdot x_{i_d} \cdot s \cdot \omega}, \\ D =& e(K, C_0^{K'} \cdot C_0') = e(g^{\frac{\alpha}{a+c}} \cdot H^r, g^{(a+c) \cdot s}) \\ =& e(g,g)^{\alpha \cdot s} \cdot e(g,h)^{(a+c) \cdot r \cdot s \cdot \omega}. \end{aligned}$$
3) Finally, recover the message as $m = \frac{C \cdot F}{D \cdot B}$.

- **KeySanityCheck** $(SK) \rightarrow (1/\perp)$. The algorithm is used to check whether a decryption key $SK = (K', K, T', L, L', L'', \{KU_i\}_{i \in path(i_d)}, \{K_\tau\}_{\tau \in I_S}, \mathcal{S})$ is well-formed. The auditor executes this algorithm as follows:
$$T', K' \in \mathbb{Z}_p, K, L, L', L'', K_\tau \in \mathbb{G}, \tag{1}$$
$$e(g, L') = e(g^a, L) \neq 1, \tag{2}$$
$$e(K, g^a \cdot g^{K'}) = e(g,g)^{\alpha} \cdot e(L^{K'} \cdot L', h^{T'}) \neq 1, \tag{3}$$
$$\exists \tau \in I_S, \text{s.t. } e(K_\tau, g) \cdot e(L \cdot L', u) = e(L'', g)^{s_\tau} \neq 1. \tag{4}$$
The decryption key SK is reviewed as a well-formed key only if it satisfies these Equations (1,2,3,4), and then the algorithm outputs 1; otherwise $\perp$.

- **Trace**$(SK_{suspected}, PP, \mathcal{R}) \rightarrow (U/authority/\perp)$. The auditor runs this algorithm. If the decryption key $SK_{suspected}$ is not well-formed, then the algorithm outputs $\perp$. Otherwise, firstly obtain $i_d = Dec_{\bar{k}}(K')$ and $T' = \omega$ from $SK_{suspected}$, and then search i_d in $LN = \{U, i_d\}$. If there not exists the same value in LN, the algorithm outputs the authority, which means that the dishonest authority fakes a user. Otherwise, the algorithm compares $T' = \omega$ in $SK_{suspected}$ with ω_U associated with the real user U. If $\omega \neq \omega_U$, the algorithm outputs the authority as a dishonest party and claim that the user is innocent. If $\omega = \omega_U$, the algorithm outputs the user U, which indicates the user U is dishonest, and generates the new revocation list $\mathcal{R}' = \mathcal{R} \bigcup \{U\}$.

- **CTUpdate**$(CT, \mathcal{R}', X') \rightarrow (CT')$. The ciphertext updated algorithm is executed by the cloud. The authority selects $\eta \in \mathbb{Z}_p$ randomly, calculates $X' = \{x_i' = \eta \cdot x_i \mod p\}_{i=0}^{2|\mathcal{U}|-2}$, and then sends them to the cloud. The update key X', the latest revocation list $\mathcal{R}'$ and the ciphertext CT are put into the algorithm by the cloud. Subsequently, the algorithm will output the new ciphertext CT' associated with the new revocation list $\mathcal{R}'$. For $j' \in cover(\mathcal{R}')$, there are two case:

1) If there exists $j \in cover(\mathcal{R})$ such that $j = j'$, then set $T_{j'} = T_j$.
2) If there exists $j \in cover(\mathcal{R})$ such that j is an ancestor of j', suppose that

$$path(j') = path(j) \bigcup \{i_{dep(j)+1}, \cdots i_{dep(j')}\},$$

where $i_{dep(j)} = j$ and $i_{dep(j')} = j'$. Let $Y_j = T_j$, compute iteratively

$$Y_{i_{k+1}} = (Y_{i_k})^{\frac{x'_{i_{k+1}}}{x'_{i_k}}} = y^s_{i_{k+1}},$$

where $k = dep(j), \cdots, dep(j') - 1$ and set $T_{j'} = Y_{j'}$. The other partial ciphertext remains to be unchanged, and then the updated ciphertext is $CT = (C, C_0, C'_0, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[1,l]}, \{T_{j'}\}_{j'\in cover(\mathcal{R}')}, \mathcal{R}', \overline{W})$.

5 Security Analysis

In this section, we first prove the IND-CPA security and accountability, and then give the proof of resistance against the collusion attacks between a revoked user and an unrevoked user.

5.1 IND-CPA Security

In the proposed AR-CP-ABE scheme, we only prove the security of the fresh ciphertext, since the distribution of updated ciphertext is as same as the fresh ciphertext. The security proof of our AR-CP-ABE scheme will be described below.

Theorem 1. *If the decisional q-BDHE hardness assumption holds, there is no polynomial time adversary that can break our AR-CP-ABE scheme with non-negligible advantage under the selective access policy and chosen plaintext attacks.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ that can break our scheme with a non-negligible advantage ε, then we can construct a challenger $\mathcal{C}$ that can solve the q-BDHE problem with the advantage $\varepsilon/2$.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplication cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and the mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Suppose that $q > 2|\mathcal{U}| - 2$. Then $\mathcal{C}$ randomly flips a fair coin $\mu = \{0, 1\}$. Given $Y' = (g, g^s, g^d, g^{d^2}, \cdots, g^{d^q}, g^{d^{q+2}}, \cdots, g^{d^{2q}})$, $\mathcal{C}$ sets $Z = e(g, g)^{d^{q+1}s}$, if $\mu = 0$; Otherwise, $\mathcal{C}$ selects $Z \in \mathbb{G}_T$ randomly. Furthermore, in order to utilize $\mathcal{A}$ to distinguish Z, $\mathcal{C}$ should simulate a challenger for $\mathcal{A}$. Thus, the simulation is as follows:

- **Initialization:** $\mathcal{A}$ chooses a challenge access policy $W^* = (M^*, \rho^*, T)$ and a revocation list $\mathcal{R}^*$, where M^* is an $l^* \times n^*$ matrix and $n^* \leq q$, ρ^* is a mapping from rows of M^* to the attribute name, and $T = \{t_{\rho^*(i)}\}_{i\in[1,l^*]}$ is the attribute value related to (M^*, ρ^*).
- **Setup:** $\mathcal{C}$ generates the public parameter as follows:
  1) Select $\alpha' \in \mathbb{Z}_p$ and set $e(g, g)^\alpha = e(g^d, g^{d^q}) \cdot e(g, g)^{\alpha'}$, which means implicitly $\alpha = \alpha' + d^{q+1}$. Then pick $a \in \mathbb{Z}_p$, compute g^a, and set $h = g^d, u = g^{d^q}$.
  2) Given the revocation list $\mathcal{R}^*$, let $I_{\mathcal{R}^*} = \{i \in path(U) | U \in \mathcal{R}^*\}$, and select $v_i \in \mathbb{Z}_p, \forall i = 0, 1, \cdots, 2|\mathcal{U}| - 2$. If $i \in I_{\mathcal{R}^*}$, set $y_i = g^{v_i} g^{d^i}$, which implies $x_i = v_i + d^i$. Otherwise, let $y_i = g^{v_i} g^{d^q}$, which means implicitly $x_i = v_i + d^q$.

  The public parameters are published as follows:

  $$PP = (p, \mathbb{G}, \mathbb{G}_T, e, g, h, u, e(g, g)^\alpha, g^a, \{y_i\}_{i=0}^{2|\mathcal{U}|-2}).$$

- **Phase 1:** To request the related intermediate keys, $\mathcal{A}$ picks randomly ω, computes $H = h^\omega$ with a zero-knowledge proof, and submits H and a series of user attribute sets $\{U, \mathcal{S} = (I_S, S)\}$ to $\mathcal{C}$, where I_S and $S = \{s_i\}_{i\in I_S}$ are the attribute name and attribute value of the user, respectively. Similar to the case in A-IBE [27], utilizing the knowledge extractor, $\mathcal{C}$ can extract ω. Then for each attribute value $s_\tau \in S$ and $i \in \{1, 2, \ldots, l^*\}$, if $s_\tau = t_{\rho^*(i)}$, set $u_\tau = s_\tau + \sum_{n=1}^{n^*} d^n M^*_{k,n}$; Otherwise, let $u_\tau = s_\tau$. According to the four combinations that whether the attribute satisfies the access policy and whether the user is revoked, $\mathcal{C}$ runs as follows:
  - **Case 1:** If $\mathcal{S} \in W^*$ and $U \notin \mathcal{R}^*$, then $\mathcal{C}$ aborts.
  - **Case 2:** If $\mathcal{S} \in W^*$ and $U \in \mathcal{R}^*$, $\mathcal{C}$ executs as follows:
    1) Choose $c \in \mathbb{Z}_p$ randomly, set $K' = c$, and compute $K, L, L', L'', \{K_\tau\}_{\tau\in I_S}$ in the followings:

$$\begin{aligned}
K &= g^{\frac{\alpha'}{a+c}} \left(g^{\frac{d^q}{a+c}}\right)^{\frac{M^*_{i,1}}{M^*_{i,2}}} = g^{\frac{\alpha}{a+c}} h^{r\omega}, \\
L &= [(g^{d^q})^{\frac{1}{(a+c)\omega}}]^{-1} [(g^{d^{q-1}})^{\frac{1}{(a+c)\omega}}]^{\frac{M^*_{i,1}}{M^*_{i,2}}} = g^r, \\
L' &= (g^r)^a, \qquad L'' = g^{r\cdot(v_{i_d} + d^{i_d})} = g^{rx_{i_d}}, \\
K_\tau &= [(g^{(v_{i_d} + d^{i_d})d^q})^{\frac{s_\tau}{(a+c)\omega}}]^{-1} \\
&\quad \cdot [(g^{(v_{i_d} + d^{i_d})d^{q-1}})^{\frac{s_\tau}{(a+c)\omega}}]^{\frac{M^*_{i,1}}{M^*_{i,2}}} \\
&\quad \cdot (g^{d^{2q}})^{\frac{1}{\omega}} [(g^{d^{2q-1}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}]^{-\frac{1}{\omega}} \\
&\quad \cdot \Big[(\prod_{k=2}^{n^*} g^{d^{q+k} M^*_{i,k}})^{-1} \\
&\quad \cdot (\prod_{k=1,k\neq 2}^{n^*} g^{d^{q+k-1} M^*_{i,k}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}\Big]^{\frac{1}{(a+c)\omega}} \\
&= g^{x_{i_d} u_\tau r} u^{-(a+c)r},
\end{aligned}$$

which means implicity $r = -\frac{d^q}{\omega(a+c)} + \frac{d^{q-1}}{\omega(a+c)} \cdot \frac{M^*_{i,1}}{M^*_{i,2}}$.

2) Suppose that $path(i_d) = \{i_0, \cdots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node related to the user U. Since $U \in \mathcal{R}^*$, then $i_d \in I_{\mathcal{R}^*}$ and $x_{i_d} = v_{i_d} + d^{i_d}$. For $i \in path(i_d)$, $\mathcal{C}$ can compute

$$\begin{aligned} KU_i =& \Big[(g^{d^q})^{-1} \cdot (g^{d^{q-1}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}\Big]^{\frac{(v_{i_d}+d^{i_d})}{(v_i+d^i)(a+c)\omega}} \\ =& g^{r\frac{x_{i_d}}{x_i}}. \end{aligned}$$

– **Case 3:** If $\mathcal{S} \notin W^*$ and $U \in \mathcal{R}^*$, $\mathcal{C}$ does as follows:

1) Select $\vec{\omega} = (\omega_1, \cdots, \omega_{n^*}) \in \mathbb{Z}_p^{n^*}$, where $\omega_1 = -1$ and $M_i^* \cdot \vec{\omega} = 0$ for all i such that $\rho^*(i) \in I_S$. Select $c \in \mathbb{Z}_p$ randomly, and set $K' = c$.

2) Choose $t \in \mathbb{Z}_p$ randomly, and calculate K, L, L', L'':

$$\begin{aligned} K =& (g^{\alpha'+dt}\prod_{i=2}^{n^*} g^{\omega_i d^{q+2-i}})^{\frac{1}{a+c}} = g^{\frac{\alpha}{a+c}} h^{r\omega}, \\ L =& [g^{\frac{t}{a+c}} \prod_{i=1}^{n^*} (g^{\omega_i d^{q+1-i}})^{\frac{1}{a+c}}]^{\frac{1}{\omega}} = g^r, \\ L' =& (g^r)^a, \qquad L'' = g^{r\cdot(v_{i_d}+d^{i_d})} = g^{r x_{i_d}}, \end{aligned}$$

which means implicity $r = \frac{1}{(a+c)\omega}(t+\omega_1 d^q + \omega_2 d^{q-1} + \cdots + \omega_{n^*} d^{q-n^*+1})$.

3) $\forall \tau \in I_S$, if $\exists i, s.t.\ \rho^*(i) = \tau$ and $s_\tau = t_{\rho^*(i)}$, then $\mathcal{C}$ computes K_τ as follows:

$$\begin{aligned} K_\tau =& \Big[\prod_{j=1}^{n^*} (g^{td^j} \prod_{k=1}^{n^*} g^{\omega_k d^{q+1+j-k}})^{M_{i,j}}\Big]^{\frac{1}{(a+c)\omega}} \\ & \cdot \Big(g^{td^q} \prod_{i=1}^{n^*} g^{\omega_i d^{2q+1-i}}\Big)^{-\frac{1}{\omega}} \cdot L^{(v_{i_d}+d^{i_d})s_\tau} \\ =& g^{x_{i_d} u_\tau r} u^{-(a+c)r}. \end{aligned}$$

Otherwise, $\mathcal{C}$ computes K_τ as follows: $K_\tau = L^{(v_{i_d}+d^{i_d})s_\tau} (g^{td^q} \prod_{i=1}^{n^*} g^{\omega_i d^{2q+1-i}})^{-\frac{1}{\omega}}$.

4) Suppose that $path(i_d) = \{i_0, \cdots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node related to the user U. Since $U \in \mathcal{R}^*$, $i_d \in I_{\mathcal{R}^*}$ and $x_{i_d} = v_{i_d} + d^{i_d}$. For $i \in path(i_d)$, $\mathcal{C}$ computes

$$KU_i = \Big(g^t \prod_{i=1}^{n^*} g^{\omega_i d^{q+1-i}}\Big)^{\frac{(v_{i_d}+d^{i_d})}{(v_i+d^i)(a+c)\omega}} = g^{r\frac{x_{i_d}}{x_i}}.$$

– **Case 4:** If $\mathcal{S} \in W^*$ and $U \notin \mathcal{R}^*$, then $K, K', L, L', L'', \{K_\tau\}_{\tau \in I_S}$ can be calculated as Case 3. Since $U \notin \mathcal{R}^*$, then $i_d \notin I_{\mathcal{R}^*}$ and $x_{i_d} = v_{i_d} + d^q$. Next, for $i \in path(i_d)$, $\mathcal{C}$ sets

$$KU_i = \Big(g^t \prod_{i=1}^{n^*} g^{\omega_i d^{q+1-i}}\Big)^{\frac{(v_{i_d}+d^q)}{(v_i+d^q)(a+c)\omega}} = g^{r\cdot x_{i_d}}.$$

- **Challenge:** $\mathcal{A}$ sends two equal-length messages m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ computes a challenge ciphertext as follows:

1) $\mathcal{C}$ flips a fair coin $v \in \{0,1\}$ and computes $C = m_v \cdot Z \cdot e(g,g)^{\alpha' s}, C_0 = g^s, C_0' = (g^a)^s$.

2) $\mathcal{C}$ selects $r_2, \cdots, r_{n^*} \in \mathbb{Z}_p^*$ randomly, sets $\vec{v} = (s, sd + r_2, \cdots, sd^{n^*-1} + r_{n^*})^\top \in \mathbb{Z}_p^{n^*}$, and then computes

$$\begin{aligned} C_{i,1} =& \prod_{j=2}^{n^*} (g^{dr_j})^{M^*_{i,j}} \prod_{j=1}^{n^*} (g^{sd^j})^{M^*_{i,j}} g^{-ad^{q+i}}, \\ C_{i,2} =& (g^{t_{\rho^*(i)}})^{-ad^i} \prod_{j=2}^{n^*} (g^{d^j M^*_{i,j}})^{-ad^i} \prod_{j=2}^{n^*} (g^{r_j})^{M^*_{i,j}} \\ & \cdot \prod_{j=1}^{n^*} (g^{sd^{j-1}})^{M^*_{i,j}} = g^{-t_i u_{\rho^*(i)} + \lambda_i}, \\ C_{i,3} =& g^{-ad^i}. \end{aligned}$$

3) $\forall j \in cover(\mathcal{R}^*)$, since $x_j = v_j + d^q$ and $y_j = g^{v_j + d^q}$, then $\mathcal{C}$ sets $T_j = (g^s)^{v_j + d^q} = y_j^s$.

Finally, $\mathcal{C}$ sends the challenge ciphertext $CT = (C, C_0, C_0^{'}, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [1,l^*]}, \{T_j\}_{j \in cover(\mathcal{R}^*)})$ to $\mathcal{A}$.

- **Phase 2:** Phase 2 is the same as Phase 1.

- **Guess:** A guess v' of v will be output by $\mathcal{A}$.

1) If $v = v'$, $\mathcal{C}$ will output a guess $\mu' = 0$ of μ. In this case, $\mathcal{C}$ sets $Z = e(g,g)^{d^{q+1}}$ and $\mathcal{A}$ will obtain a legal ciphertext. Since the advantage of $\mathcal{A}$ is ε, $|\Pr[v = v'|\mu = 0] - \frac{1}{2}| = \varepsilon$. Furthermore, $\Pr[v = v'|\mu = 0] = \Pr[\mu = \mu'|\mu = 0]$ can be concluded. Then we have $\Pr[\mu = \mu'|\mu = 0] = \varepsilon + \frac{1}{2}$.

2) If $v \neq v'$, $\mathcal{C}$ outputs a guess $\mu' = 1$ of μ. In this case, $\mathcal{C}$ selects $Z \in \mathbb{G}_T$ randomly and $\mathcal{A}$ cannot obtain any information of v. Thus, the advantage of $\mathcal{A}$ is $\frac{1}{2}$, that is to say, $\Pr[v \neq v'|\mu = 1] = \frac{1}{2}$, In addition, $\Pr[v \neq v'|\mu = 1] = \Pr[\mu = \mu'|\mu = 1]$ is easily concluded. Therefore, we have $\Pr[\mu = \mu'|\mu = 1] = \frac{1}{2}$.

Finally, the advantage of $\mathcal{C}$ in the game is

$$\begin{aligned}|\Pr[\mu = \mu'] - \frac{1}{2}| =& |\Pr[\mu = \mu'|\mu = 0]\Pr[\mu = 0] \\ &+ \Pr[\mu = \mu'|\mu = 1]\Pr[\mu = 1] - \frac{1}{2}| \\ =& |(\varepsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2}| \\ =& \frac{1}{2}\varepsilon.\end{aligned}$$

□

5.2 Accountability Security

In this section, we prove the accountability security of AR-CP-ABE scheme by the following three theorems.

Theorem 2. *If the DLP hardness assumption holds, the advantage of an adversary in the Dishonest-Authority game is negligible for AR-CP-ABE scheme.*

Proof. Assume that there exists a PPT adversary $\mathcal{A}$ who has a non-negligible advantage ε in the Dishonest-Authority game, then we can construct a challenger $\mathcal{C}$ that can solve a DLP problem with a non-negligible advantage ε.

Furthermore, in order to utilize $\mathcal{A}$ to solve the DLP problem, $\mathcal{C}$ should interact with $\mathcal{A}$ as follows:

- **Setup:** The adversary $\mathcal{A}$ calls the Setup algorithm and submits the public parameters $PP = (p, \mathbb{G}, \mathbb{G}_T, e, g, h, u, e(g,g)^\alpha, g^a, \{y_i\}_{i=0}^{2|\mathcal{U}|-2})$ and $c = E_{\bar{k}}(i_d)$ about a user U to $\mathcal{C}$.

- **Key Query:** $\mathcal{C}$ receives a challenge $H = h^\omega \in \mathbb{G}$, where ω is unknown for $\mathcal{C}$. Using rewinding techniques of Zero-knowledge Proof of Knowledge of Discrete log protocol in Goyal's scheme [27], $\mathcal{C}$ can give the required proof without knowledge of ω. Then $\mathcal{A}$ computes $(K', K, L, L', L'', \{KU_i\}_{i\in path(i_d)}, \{K_\tau\}_{\tau\in I_S}, \mathcal{S})$ to $\mathcal{C}$.

- **Key Forgery:** $\mathcal{A}$ outputs a decryption key $SK^* = ((K')^*, K^*, (T')^* = \omega', L^*, (L')^*, (L'')^*, \{K_\tau^*\}_{\tau\in I_S}, \{KU_i^*\}_{i\in path(i_d)}, \mathcal{S})$ associated with U. Then $T' = \omega'$ will be a solution of the discrete logarithm problem if SK^* is well-formed.

If $\mathcal{A}$ can successfully forge a decryption key, $\mathcal{C}$ must solve the discrete logarithm problem. Since DLP hardness assumption cannot be solved in probabilistic polynomial time, there does not exist an $\mathcal{A}$ who has a non-negligible advantage in the *Dishonest-Authority* game. □

Theorem 3. *If the l-SDH hardness assumption holds, the advantage of an adversary in the Dishonest-User-1 game is negligible for the AR-CP-ABE scheme under $q < l$, where q is the number of key query.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ who has a non-negligible advantage ε in the *Dishonest-User-1* game with q key queries and $l = q + 1$, then we can construct a challenger $\mathcal{C}$ that attacks the l-SDH hardness assumption with a non-negligible advantage ε. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplication cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear mapping. Given an l-SDH problem $(g_1, g_1^a, g_1^{a^2}, \cdots, g_1^{a^l})$, where $g_1 \in \mathbb{G}, a \in \mathbb{Z}_p$, the objective of $\mathcal{C}$ is to find a tuple $(c_r, \varpi_r = g_1^{\frac{1}{a+c_r}})$. For $i = 0, 1, \cdots, l$, set $A_i = g_1^{a^i}$. Then the simulation will be executed as follows:

- **Setup:** $\mathcal{C}$ selects randomly q different values $c_1, c_2, \cdots, c_q \in \mathbb{Z}_q^*$ and $\alpha, \theta \in \mathbb{Z}_p, u \in \mathbb{G}$. Let $f(y) = \prod_{i=1}^{q}(y + c_i) = \sum_{i=0}^{q} \alpha_i y^i$, where $\alpha_0, \cdots, \alpha_q \in \mathbb{Z}_p$ are the coefficients of $f(y)$. Then $\mathcal{C}$ executes as follows:

    1) Let $g = \prod_{i=0}^{q}(A_i)^{\alpha_i} = g_1^{f(a)}$, $g^a = \prod_{i=1}^{q+1}(A_i)^{\alpha_{i-1}} = g_1^{f(a)\cdot a}$.

    2) For each node of $\mathcal{T}$, $\mathcal{C}$ chooses $\{x_i\}_{i=0}^{2|\mathcal{U}|-2} \in \mathbb{Z}_p$ randomly, computes $\{y_i = g^{x_i}\}_{i=0}^{2|\mathcal{U}|-2}$ and publishes the public parameters $PP =(p$, $\mathbb{G}$, $\mathbb{G}_T$, e, g, $h = g^\theta$, u, $e(g,g)^\alpha$, g^a, $\{y_i = g^{x_i}\}_{i=0}^{2|\mathcal{U}|-2})$.

- **Key Query:** $\mathcal{A}$ requests q key queries. For i-th query, $\mathcal{A}$ submits $(U_i, \mathcal{S}_i)$ and $H_i = h^{\omega_i}$ to $\mathcal{C}$, where $\omega_i \in \mathbb{Z}_p$. Then set

$$f_i(y) = \frac{f(y)}{y + c_i} = \prod_{j=1, j\neq i}^{q}(y + c_j) = \sum_{j=0, j\neq i}^{q-1} \beta_j y^j,$$

where $\beta_0, \cdots, \beta_{q-1} \in \mathbb{Z}_p$ are the coefficients of $f_i(y)$. $\mathcal{C}$ computes

$$\sigma_i = \prod_{j=0}^{q-1}(A_j)^{\beta_j} = g_1^{f_i(a)} = g_1^{\frac{f(a)}{a+c_i}} = g^{\frac{1}{a+c_i}}.$$

Then $\mathcal{C}$ chooses $r \in \mathbb{Z}_p$ randomly and computes a partial key about (U_i, S_i) as: $(K' = c_i, K = (\sigma_i)^\alpha = g^{\frac{\alpha}{a+c_i}} H^r, L = g^r, L' = (g^a)^r, L'' = g^{rx_{i_d}}, \{K_\tau = g^{x_{i_d} s_\tau r}(u^a \cdot u^{c_i})^{-r} = g^{x_{i_d} s_\tau r} u^{-(a+c_i)r}\}_{\tau\in I_S})$. Suppose $path(i_d) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is a leaf node the value associated with the user U_i in the tree. $\mathcal{C}$ sets the key component $KU_i = g^{r\cdot\frac{x_{i_d}}{x_i}}$ of the user U_i. Finally, $\mathcal{C}$ sends the intermediate key $(K', K, L, L', L'', \{K_\tau\}_{\tau\in I_S}, \{KU_i\}_{i\in path(i_d)}, \mathcal{S}_i)$ to $\mathcal{A}$.

- **Key Forgery:** $\mathcal{A}$ sends a forged key SK^* to $\mathcal{C}$. Let ξ_1 stand for the event that $\mathcal{A}$ wins the Dishonest-User-1 game. Suppose that SK^* satisfies the conditions of the key sanity check and $K' \notin \{c_1, \cdots, c_q\}$.

    1) If ξ_1 dose not occur, $\mathcal{C}$ chooses a tuple $(c_r, \varpi_r) \in \mathbb{Z}_p \times \mathbb{G}$ as the solution of l-SDH problem.

2) If ξ_1 occurs, $\mathcal{C}$ writes a polynomial $f(y) = \varphi(y)(y+K') + \varphi - 1$, where $\varphi(y) = \sum_{i=0}^{q-1} \varphi_i y^i$ and $\varphi - 1 \in \mathbb{Z}_p^*$. Since $f(y) = \prod_{i=1}^{q}(y+c_i)$, $c_i \in \mathbb{Z}_p^*$ and $K' \notin \{c_1, \cdots, c_q\}$, $(y+K')$ can not divide into $f(y)$. $\mathcal{C}$ sets $\sigma = (K/L^{\theta T'})^{\alpha^{-1}} = g^{\frac{1}{a+K'}} = g_1^{\frac{f(a)}{a+K'}} = g_1^{\varphi(a)} g_1^{\frac{\varphi-1}{a+K'}}$ and then can compute $c_r = K', \varpi_r = (\sigma \cdot \prod_{i=0}^{q-1} A_i^{-\varphi_i})^{\frac{1}{\varphi-1}} = g_1^{\frac{1}{a+K'}}$. Since $e(g_1^a g_1^{c_r}, \varpi_r) = e(g_1^a g_1^{K'}, g_1^{\frac{1}{a+K'}}) = e(g_1, g_1)$, (c_r, ϖ_r) is the solution to the l-SDH problem.

Let ξ_2 denote the event that (c_r, ϖ_r) is the solution to the l-SDH problem. If $\mathcal{C}$ randomly selects (c_r, ϖ_r), ξ_2 occurs with negligible advantage, for simplicity with 0. In the case where $\mathcal{A}$ succeeds and $\gcd(\varphi - 1, p) = 1$, the probability of (c_r, ϖ_r) that satisfies the condition $e(g_1^a g_1^{c_r}, \varpi_r) = e(g_1^a g_1^{K'}, g_1^{\frac{1}{a+K'}})$ is 1. So the probability of $\mathcal{C}$ to solve the l-SDH problem is:

$$\begin{aligned}
\Pr[\xi] =& \Pr[\xi|\overline{\mathcal{A}\ succeeds}] \cdot \Pr[\overline{\mathcal{A}\ succeeds}] \\
&+ \Pr[\xi|\mathcal{A}\ succeeds \wedge \gcd(\varphi-1,p) \neq 1] \\
&\cdot \Pr[\mathcal{A}\ succeeds \wedge \gcd(\varphi-1,p) \neq 1] \\
&+ \Pr[\xi|\mathcal{A}\ succeeds \wedge \gcd(\varphi-1,p) = 1] \\
&\cdot \Pr[\mathcal{A}\ succeeds \wedge \gcd(\varphi-1,p) = 1] \\
=& 0 + 0 + 1 \cdot \Pr[\mathcal{A}\ succeeds \wedge \gcd(\varphi-1,p) = 1] \\
=& \varepsilon.
\end{aligned}$$

□

Theorem 4. *If the DLP hardness assumption holds, the advantage of an adversary in the Dishonest-User-2 game is negligible for the AR-CP-ABE scheme.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ that has a non-negligible advantage ε in the *Dishonest-User-2* game, then we can construct a challenger $\mathcal{C}$ that can solve the DLP problem (g, g^z) with a non-negligible advantage. In order to utilize $\mathcal{A}$ to obtain z, $\mathcal{C}$ should simulate a challenger for $\mathcal{A}$, and interacts with $\mathcal{A}$ as follows:

- **Setup:** $\mathcal{C}$ sends the public parameters PP to $\mathcal{A}$ by calling the Setup algorithm. Then $\mathcal{C}$ selects $t, \mu \in \mathbb{Z}_p$ randomly and computes $h = g^t, u = g^\mu$.

- **Key Query:** $\mathcal{A}$ submits a series of attribute sets to $\mathcal{C}$ for requesting the intermediate keys. For every query, when $\mathcal{A}$ makes a proof of knowledge of the discrete log of h^ω with respect to h for $\mathcal{C}$, $\mathcal{C}$ will extract the discrete log ω by using a knowledge extractor [27]. Then $\mathcal{C}$ selects $\gamma \in \mathbb{Z}_p$ and computes $(K' = c, K = g^{\frac{\alpha}{a+c}} g^{\omega z \gamma t} = g^{\frac{\alpha}{a+c}} g^{\omega r t}, L = g^{z\gamma} = g^r, L' = g^{az\gamma} = g^{ar}, L'' = g^{x_{i_d} z\gamma} = g^{x_{i_d} r}, \{K_\tau = g^{x_{i_d} s_\tau z\gamma} g^{-(a+c)z\gamma\mu}\}_{\tau \in I_S} = g^{x_{i_d} s_\tau r} g^{-(a+c)r\mu}\}_{\tau \in I_S})$, which means implicitly $r = \gamma \cdot z$. Finally, $\mathcal{C}$ sends $(K', K, L, L', L'', \{K_\tau\}_{\tau \in I_S}, \mathcal{S})$ to $\mathcal{A}$.

- **Key Forgery:** $\mathcal{A}$ outputs a forged key SK^* related with (U, c). Suppose that (U, c) has been queried, let's call that (U_i, c_i), but ω does not equal to ω_i. The key of (U_i, c_i) is $(K' = c, K = g^{\frac{\alpha}{a+c}} h^{\omega r}, L = g^r, L' = g^{ar}, L'' = g^{x_{i_d} r}, \{K_\tau = g^{x_{i_d} s_\tau r} u^{-(a+c)r}\}_{\tau \in I_S}, T' = \omega_i)$. $\mathcal{A}$ outputs a forged key $SK^* = \big((K')^* = c, K^* = g^{\frac{\alpha}{a+c}} h^{\omega^* r^*}, L^* = g^{r^*}, (L')^* = g^{ar^*}, (L'')^* = g^{x_{i_d} r^*}, \{K_\tau^* = g^{x_{i_d} s_\tau r^*} u^{-(a+c)r^*}\}_{\tau \in I_S}, (T')^* = \omega^*\big)$.

Now, we analyze K and K^*, K_τ and K_τ^*. If $\mathcal{A}$ can forge K^* and K_τ^* successfully, then we can suppose that $K^* = K \cdot h^{p_1} \Rightarrow \omega r + p_1 = \omega^* r^*$ and $K_\tau^* = K_\tau^{p_2} \Rightarrow r p_2 = r^*$. Since $\mathcal{A}$ knows $\omega, \omega^*, p_1, p_2$, then $\mathcal{A}$ can get $r = p_1/(\omega^* p_2 - \omega)$. Suppose that the probability $\omega^* p_2 = \omega$ can be negligible and since $r = \gamma z$, $\mathcal{A}$ can compute the solution of DLP problem $z = r/\gamma = p_1/\gamma(\omega^* p_2 - \omega)$. However DLP hardness assumption cannot be solved in probabilistic polynomial time, there does not exist an adversary $\mathcal{A}$ who has a non-negligible advantage in the *Dishonest-User-2* game. □

5.3 Collusion Resistance

Theorem 5. *If the DLP difficulty problem holds, the proposed AR-CP-ABE scheme with user revocation is secure against user collusion in the selective model.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ who can break the proposed scheme with a non-negligible advantage ε after q_1 Type-I queries and q_2 Type-II queries, then we can construct a challenger $\mathcal{C}$ that can solve the DLP problem (g, g^z) with the advantage at most $\varepsilon/(q_1 \cdot q_2)$. Furthermore, in order to utilize $\mathcal{A}$ to obtain z, $\mathcal{C}$ should simulate a challenger for $\mathcal{A}$. Then, $\mathcal{C}$ interacts with $\mathcal{A}$ as follows:

- **Initialization:** $\mathcal{A}$ chooses a challenge access policy $W^* = (M^*, \rho^*, T^*)$ and a revocation list $\mathcal{R}^*$, where M^* is an $l^* \times n^*$ matrix and $n^* \leq q$, ρ^* is a mapping from rows of M^* to the attribute name, and $T^* = \{t_{\rho^*(i)}\}_{i \in [1, l^*]}$ is the attribute value related to (M^*, ρ^*).

- **Setup:** $\mathcal{C}$ generates the public parameters by calling the Setup algorithm and sends the public parameters PP to $\mathcal{A}$. Note that for each node of the binary tree $\mathcal{T}$, select $\{x_i\}_{i=0}^{2|\mathcal{U}|-2} \in \mathbb{Z}_p^*$ randomly. If a user $U \notin \mathcal{R}^*$, $\mathcal{C}$ sets $A = g^z$ and computes $\{y_i = A^{x_i} = g^{zx_i}\}_{i \in path(i_d)}$. Otherwise, $\mathcal{C}$ computes $\{y_i = g^{x_i}\}_{i \in path(i_d)}$.

- **Phase 1:** $\mathcal{C}$ first sets two empty lists L_I and L_{II}. $\mathcal{A}$ submits some queries as follows.

  - **Type-I key query** $\langle U_I, \mathcal{S}_I \rangle$: User U_I already has been revoked, but her or his attribute set $\mathcal{S}_I = (I_S, S) \in W^*$, where I_S and $S = \{s_i\}_{i \in I_S}$ are the attribute name and attribute value of the user. Firstly, $\mathcal{A}$ computes $H = h^\omega$ and

Table 1: Functionality comparisons

schemes	Revocation	Update	Collusion Resistance	Authority Accountability	Hidden Policy	Backward Security
Li *et al.* [11]	×	×	–	√	√	–
Vaanchig *et al.* [29]	attribute	ciphertext, key	√	×	×	√
ATIR-CPABE [16]	user	key	√	√	×	×
Han *et al.* [17]	user	ciphertext	×	×	√	×
Zhang *et al.* [18]	×	×	–	×	√	–
Ours	user	ciphertext	√	√	√	√

Table 2: Efficiency comparisons

schemes	KeyGen	Encrypt	Decrypt	Trace	Update
Li *et al.* [11]	$(7+3s)E+(1+2s)M$	$(4+5l)E$ $+(3+2l)M$	$(1+3l)P+5E$ $+(5+2l)M$	–	–
Vaanchig *et al.* [29]	$(1+s)E+M$	$(1+5l)E$ $+M$	$(2n+1)P+nE$ $+(2+n)M$	–	$2n_cE$
ATIR-CPABE [16]	$(10+s)E+(9+s)M$	$(6+3l)E$ $+(1+l)M$	$(5+2n)P+(6+n)E$ $+(6+n)M$	$(8+2s)P+7E$ $+4M$	$4E+3M$
Han *et al.* [17]	$(6+s)E+(1+s)M$	$(3+4l+r)E$ $+(1+l)M$	$(2+3n)P+(3+n)E$ $+(5+2n)M$	$(6+s)P+(2+s)E$ $+(3+s)M$	t_1E
Zhang *et al.* [18]	$(3+2s)E+(3+2s)$	$(4+6l)E$ $+(2+6l)M$	$(2n+1)P+nE+nM$	–	–
Ours	$(6+s+j)E+(1+s)M$	$(3+4l+r)E$ $+(1+l)M$	$(2+3n)P+(4+n)E$ $+(5+2n)M$	$(6+s)P+(3+s)E$ $+(3+s)M$	t_1E

An exponent operation in $\mathbb{G}_T, \mathbb{G}$ is represented by E. A bilinear pairing operation is represented by P. A multiplication is represented by M. The number of attributes that the access policy contains is represented by l. The number of attributes that the user owns is represented by s. The number of attributes that meets the access policy is represented by n. The number of $cover(R)$ is represented by r. The length of $path(U)$ is represented by j.

gives a zero-knowledge proof to $\mathcal{C}$ for requesting the intermediate keys. In Goyal's scheme [27], a simulator can use a knowledge extractor to extract ω. Thus $\mathcal{C}$ can use this technology to obtain ω. Then $\mathcal{C}$ can generate a intermediate key in the following:

1) Choose $c, r_I \in \mathbb{Z}_p$ randomly, and compute $K', K, L, L', L'', \{K_\tau\}_{\tau \in I_S}$ as follows:

$$\begin{aligned} K' &= c, & K &= g^{\frac{\alpha}{a+c}} h^{r_I \omega}, \\ L &= g^{r_I}, & L' &= (g^{r_I})^a = g^{ar_I}, \\ L'' &= g^{r_I x_{I,i_d}}, & K_\tau &= g^{x_{I,i_d} \cdot s_\tau \cdot r_I} \cdot u^{-(a+c)\cdot r_I}, \end{aligned}$$

where $\tau \in I_S$.

2) Suppose that $path(i_d) = \{i_0, \cdots, i_d\}$, where $i_0 = root$ and i_d is the value of the leaf node related to the user U_I. $\mathcal{C}$ computes $\{KU_{I,i} = g^{r_I \cdot \frac{x_{I,i_d}}{x_{I,i}}}\}_{i \in path(i_d)}$. Finally, $\mathcal{C}$ sends the key $(K', K, L, L', L'', \{KU_{I,i}\}_{i \in path(i_d)}, \{K_\tau\}_{\tau \in I_S}, \mathcal{S})$ to $\mathcal{A}$ and adds it into L_I.

– **Type-II key query** $\langle U_{II}, \mathcal{S}_{II} \rangle$: User U_{II} is unrevoked, but her or his attribute set $\mathcal{S}_{II} \notin W^*$. Then $\mathcal{C}$ chooses $c, r_{II} \in \mathbb{Z}_p$ randomly and generates a intermediate key by running the KeyGen algorithm as follows.

$$\begin{aligned} K' &= c, \qquad K = g^{\frac{\alpha}{a+c}} \cdot h^{r_{II}\cdot\omega}, \qquad L = g^{r_{II}}, \\ L' &= g^{a \cdot r_{II}}, \qquad L'' = (g^z)^{r_{II} \cdot x_{II,i_d}}, \\ K_\tau &= (g^z)^{x_{II,i_d} \cdot s_\tau \cdot r_{II}} \cdot u^{-(a+c)\cdot r_{II}}, \\ \{KU_{II,i} &= g^{r_{II} \cdot \frac{x_{II,i_d}}{x_{II,i}}}\}_{i \in path(i_d)}. \end{aligned}$$

Finally, $(K', K, L, L', L'', \{KU_{II,i}\}_{i \in path(i_d)}, \{K_\tau\}_{\tau \in I_S}, \mathcal{S})$ will be sent to $\mathcal{A}$ and added to L_{II} by $\mathcal{C}$.

- **Challenge:** $\mathcal{A}$ sends two equal-length messages m_0, m_1 to $\mathcal{C}$. Then $\mathcal{C}$ flips a coin $\bar{b} \in \{0,1\}$ randomly and computes a ciphertext of $m_{\bar{b}}$ as follows:

  1) Select $k_i, s \in \mathbb{Z}_p$ randomly, where $i \in [1, l]$, and calculate a partial ciphertext encrypted by the access policy W^*: $(C = m_{\bar{b}} \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{as}, \{C_{i,1} = h^{\lambda_i} u^{k_i}, C_{i,2} = g^{-k_i \cdot t_{\rho(i)} + \lambda_i}, C_{i,3} = g^{k_i}\}_{i \in [1,l]})$.

  2) Set the other ciphertext component $(\{T_j = y_j^s =$

$g^{zx_j s}\}_{j \in cover(\mathcal{R}^*)})$, which is related to the revocation list $\mathcal{R}^*$.

- **Phase 2:** Phase 2 is as same as Phase 1.
- **Guess:** If the challenge ciphertext can be decrypted by $\mathcal{A}$, he has to combine $K', K, L, L', L'', \{K_\tau\}_{\tau \in I_S}$ of U_I and $\{KU_{II,i}\}_{i \in path(i_d)}$ of U_{II}. Then $\mathcal{C}$ can successfully select matching tuples $K', K, L, L', L'', \{K_\tau\}_{\tau \in I_S}$ and $\{KU_{II,i}\}_{i \in path(i_d)}$ from L_I and L_{II}. Hence, he can compute

$$\begin{aligned} B =& e(KU_{II,j}, T_j)^{T'} = e(g^{r_{II} \cdot \frac{x_{II,i_d}}{x_j}}, y_j^s)^\omega \\ =& e(g^{r_{II} \cdot \frac{x_{II,i_d}}{x_j}}, g^{zx_j s})^\omega = e(g,g)^{r_I \cdot x_{I,i_d} \cdot s \cdot \omega}. \end{aligned}$$

  Therefore, only if $r_{II} \cdot x_{II,i_d} \cdot z = r_I \cdot x_{I,i_d}$, the above equation holds and the ciphertext can be decrypted correctly. Finally, $\mathcal{C}$ outputs $z = \frac{r_I \cdot x_{I,i_d}}{r_{II} \cdot x_{II,i_d}}$ as his answer.

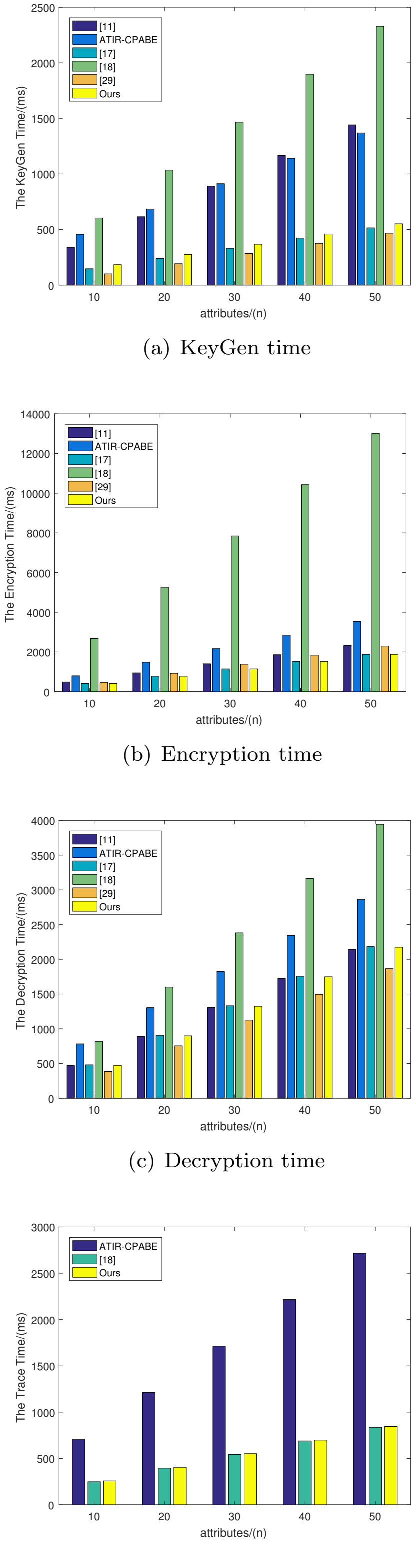
Suppose that $\mathcal{A}$ issues q_1 Type-I key queries and q_2 Type-II key queries, then the probability that $\mathcal{C}$ selects matching tuples is $1/(q_1 \cdot q_2)$. Thus the advantage of $\mathcal{C}$ is at most $\varepsilon/(q_1 \cdot q_2)$. □

6 Performance Analysis

In this section, we will compare the functionality and evaluate the efficiency between the proposed scheme and the existing schemes [11, 16–18, 29].

6.1 Functionality Comparisons

Table 1 shows that Ning *et al.*'s implicitly revocable CP-ABE scheme (ATIR-CPABE) [16] and the proposed AR-CP-ABE scheme can support the authority and user accountability and user revocation, while Han *et al.*'s scheme [17] and Zhang *et al.*'s [18] scheme can achieve hidden policy merely. Vaanchig *et al.* [29] can also implement user revocation and Li *et al.*'s scheme [11] can also realize the authority accountability. However, Vaanchig *et al.* [29] cannot trace the malicious user and Li *et al.*'s scheme [11] cannot remove the malicious users from the E-health system. Furthermore, Ning *et al.*'s scheme [16] cannot implement the backward security and the hidden policy, Han *et al.*'s scheme [17] cannot support the accountability and the backward security. At the same time, Ning *et al.*'s [16] and Han *et al.*'s [17] schemes could be vulnerable to user collusion attacks. Fortunately, the proposed AR-CP-ABE scheme can implement these functionalities at the same time and avoid the above security flaws.



(a) KeyGen time

(b) Encryption time

(c) Decryption time

(d) Trace time

Figure 3: The comparisons of the results

6.2 Efficiency Analysis

In Table 2, we denote $t_1 = \sum_{j' \in cover(R')} (dep(j') - 1 - dep(j))$ and n_c as the number of ciphertexts including an attribute in its access structure. Table 2 shows that in the KeyGen and Trace algorithm, since the pairing operation takes more time than the exponent operation, the proposed scheme is more efficient than Li *et al.*'s scheme [11] and Ning *et al.*'s ATIR-CPABE scheme [16] in Trace al-

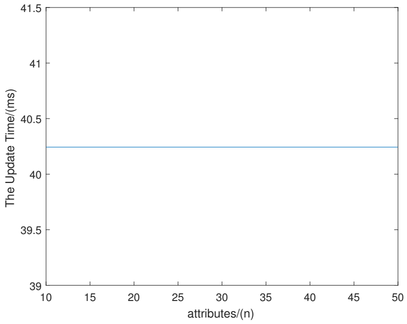


Figure 4: The efficiency of the update

gorithm. In the update algorithm, ATIR-CPABE scheme [16] and Vaanchig *et al.* [29] needs to generate the updated keys for all the unrevoked users, while only the ciphertext needs to be updated in the proposed scheme, thus the update time cannot be compared. It is pointed out that the proposed scheme can implement the accountability, user revocation and policy hiding at the same time, but the efficiency of the proposed scheme is comparable to that of Han *et al.*'s scheme [17].

Furthermore, Figure 3 demonstrates the efficiency test about the proposed scheme and related schemes and Figure 4 shows the efficiency of the Update algorithm with the number of attributes from 10 to 50. The machine for execution is 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz with 16.0GB RAM running 64 bits Windows 10. We set the attribute number from 10 to 50 and the revocation list from $\mathcal{R} = \varnothing$ to $\mathcal{R}^* = \{U_6\}$. Figure 3 vividly shows the comparisons of the KeyGen time, the Encryption time, Decryption time and the Trace time, respectively. It is clear that the proposed scheme is more efficient than other schemes [11, 16, 18]. Figure 4 shows that the update time is independent of the attributes in our scheme, which only updates the ciphertexts.

7 Conclusions

In this paper, we have presented a collusion resistance CP-ABE scheme with accountability, revocation and policy hiding. By binding together the secret value of the binary tree decryption node to the specific information of users, the proposed scheme can avoid user collusion attacks and achieve the backward security. At the same time, our scheme can implement the white-box accountability by embedding the secret value of user in the key and the partial hidden policy. Furthermore, the proposed scheme is proved to be secure under the decisional q-BDHE hardness assumption in the standard model. In the future, we aim to construct an accountable and revocable CP-ABE scheme with full hidden policy and use fine-grained attribute revocation to manage user permissions.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants No.61807026, the National Key R&D Program of China under Grant No.2017YFB0802000, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No.2019JM-198, the Plan For Scientific Innovation Talent of Henan Province under Grant No.184100510012, and in part by the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant No.18HASTIT022.

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption." The Advances in Cryptology-EUROCRYPT, Springer, 2005, pp. 457-473.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption." The 2007 IEEE Symposium on Security and Privacy, IEEE, 2007, pp. 321-334.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data." The 13th ACM Conference on Computer and Communications Security, ACM, 2006, pp. 89-98.

[4] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zhang, "Attribute-based encryption for cloud computing access control: a survey." ACM Computing Surveys, Vol. 53, no. 4, pp. 83:1-83:41, 2020.

[5] Z. Liu, Z. Cao , and D. S. Wong, "Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on eBay." The 2013 ACM SIGSAC Conference on Computer and Communications, ACM, 2013, pp. 475-486.

[6] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability." The 12th International Conference on Information Security, Springer, 2009, pp. 347-362.

[7] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures." IEEE Transactions on Information Forensics and Security, Vol. 8, no. 1, pp.76-88, 2013.

[8] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes." IEEE Transactions on Information Forensics and Security, Vol. 10, no. 6, pp.1274-1288, 2015.

[9] J. Ning, X. Dong, and Z. Cao, "Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud." The 20th European Symposium on Research in Computer Security, Springer, 2015, pp. 270-289.

[10] Y. Zhang, J. Li, D. Zheng, X. Chen, and H. Li, "Towards privacy protection and malicious behavior

traceability in smart health." Personal and Ubiquitous Computing, Vol. 21, no. 5, pp. 851-830, 2017.

[11] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for cloudIoT." IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2020.2975184.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation." The 5th ACM Symposium on Information, Computer and Communications Security, ACM, 2010, pp. 261-270.

[13] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems." IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, 2011.

[14] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage." IEEE Systems Journal, Vol. 12, no. 2, pp. 1767-1777, 2018.

[15] K. Lee, S. G. Choi, D. H. Lee,J. H. Park, and M. Yung, "Self-updatable encryption: time constrained access control with hidden attributes and better efficiency." The Advances in Cryptology-ASIACRYPT, Springer, 2013, pp. 235-254.

[16] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K. K. R. Choo, "CryptCloud+: secure and expressive data access control for cloud storage." IEEE Transactions on Services Computing, Vol. 14, no. 1, pp. 111-124, 2021.

[17] D. Han, N. Pan, and K. C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection." IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2020.2977646.

[18] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attribute-based access control." IEEE Internet of Things Journal, Vol. 5, no. 3, pp. 2130-2145, 2018.

[19] L. Sun, and C. Xu, "Hidden policy ciphertext-policy attribute based encryption with conjunctive keyword search." The 3rd IEEE International Conference on Computer and Communications, IEEE, 2017, pp. 1439-1443.

[20] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers." The Advances in Cryptology-CRYPTO, Springer, 2001, pp. 41-62.

[21] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization." The Public Key Cryptography-PKC, Springer, 2011, pp. 53-70.

[22] D. Boneh andk B. Xavier, "Short signatures without random oracles." The Advances in Cryptology-EUROCRYPT, Springer, 2004, pp. 56-73.

[23] B. Smith, "Isogenies and the discrete logarithm problem in jacobians of genus 3 hyperelliptic curves." The Advances in Cryptology-EUROCRYPT, Springer, 2008, pp. 163-180.

[24] M. Bahrami, and M. Singhal, "A dynamic cloud computing platform for eHealth systems." The 17th International Conference on E-health Networking, Application and Services (HealthCom), IEEE, 2015, pp. 435-438

[25] Z. Liu, F. Yin, J. Ji, and B. Wang, "Revocable and searchable attribute-based encryption scheme with multi-keyword and verifiability for internet of things." International Journal of Network Security, Vol. 23, no. 2, pp. 205-219, 2021.

[26] Z. Liu, Y. Liu, J. Xu, and B. Wang, "Verifiable attribute-based keyword search encryption with attribute revocation for electronic health record system." International Journal of Network Security, Vol. 22, no. 5, pp. 845-856, 2020.

[27] V. Goyal, "Reducing trust in the PKG in identity based cryptosystem." The Advances in Cryptology-CRYPTO, Springer, 2007, pp. 430-447.

[28] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service." Journal of Circuits Systems and Computers, Vol 26, No. 5, 1750072: 1-16, 2016.

[29] N Vaanchig, H. Xiong, W. Chen, and Z. Qin, "Achieving collaborative cloud data storage by key-escrow-free multi-authority CP-ABE scheme with dual-revocation." International Journal of Network Security, Vol 20, No.1, PP. 95-109, 2018.

Biography

Zhenhua Liu received the B.S. degree from Henan Normal University in 2000, and the M.S. and Ph.D. degrees from Xidian University, China, in 2003 and 2009, respectively. He is currently a Professor of Xidian University, China. His current research interests include cryptography and information security.

Yingying Ding received the B.S. degree from Henan Normal University in 2019. She is currently going in for the M.S. degree in mathematics with Xidian University, China. Her research interests concentrate on cryptography and cloud security.

Ming Yuan received the B.S. degree from Henan Normal University in 2018. She is currently going in for the M.S. degree in mathematics with Xidian University, China. Her research focuses on network and information security .

Baocang Wang received the B.S., the M.S. and Ph.D. degrees from Xidian University, China, in 2001, 2004, and 2006, respectively. He is currently a Professor with the State Key Laboratory of Integrated Services Networks of Xidian University, China. His research focuses on post-quantum cryptography, number theoretic algorithms, and cloud security.