Accountable Attribute-based Encryption with Public Auditing and User Revocation in the Personal Health Record System

Wei Zhang, Yi Wu, Hu Xiong, and Zhiguang Qin[*]
University of Electronic Science and Technology of China
Chengdu, 610054, China
[e-mail: qinzg@uestc.edu.cn, 784143121@qq.com, xionghu.uestc@gmail.com, qinzg@uestc.edu.cn]
*Corresponding author: Zhiguang Qin

Received October 17, 2020; revised December 18, 2020; accepted January 13, 2021;
published January 31, 2021

Abstract

In the system of ciphertext policy attribute-based encryption (CP-ABE), only when the attributes of data user meets the access structure established by the encrypter, the data user can perform decryption operation. So CP-ABE has been widely used in personal health record system (PHR). However, the problem of key abuse consists in the CP-ABE system. The semi-trusted authority or the authorized user to access the system may disclose the key because of personal interests, resulting in illegal users accessing the system. Consequently, aiming at two kinds of existing key abuse problems: (1) semi-trusted authority redistributes keys to unauthorized users, (2) authorized users disclose keys to unauthorized users, we put forward a CP-ABE scheme that has authority accountability, user traceability and supports arbitrary monotonous access structures. Specifically, we employ an auditor to make a fair ruling on the malicious behavior of users. Besides, to solve the problem of user leaving from the system, we use an indirect revocation method based on trust tree to implement user revocation. Compared with other existing schemes, we found that our solution achieved user revocation at an acceptable time cost. In addition, our scheme is proved to be fully secure in the standard model.

Keywords: ABE, Personal Health Record, Accountable, Public Auditing, Revocable

http://doi.org/10.3837/tiis.2021.01.017 **ISSN : 1976-7277**

1. Introduction

Due to the high flexibility and scalability of cloud computing, many businesses and individuals rely on cloud servers to store and calculate their data [1-5]. Today, cloud computing technology is relatively mature and widely used. In this way, they can not only save money, but also improve efficiency. The cloud server is not completely trusted, so the data now placed on the cloud are all in ciphertext [2, 6]. Take a PHR system as an example, patients upload their own personal medical records to the PHR system, through which doctors know the patient's medical records and make a rapid diagnosis. This method not only saves patients' time and money, but also improves the efficiency of diagnosis for doctors. A PHR system based on traditional encryption scheme is shown in **Fig. 1**. However, the traditional encryption scheme can not provide fine-grained access control, which greatly limits the application of PHR system [4, 7]. Therefore, many secure PHR systems [8-12] are built based on attribute-based encryption (ABE). ABE is regarded as the most compelling encryption primitive that realizes fine-grained [1, 13] access control, solving the problem of one-to-many secret data sharing. In ABE, users have a series of attributes to identify themselves. Only if these attributes they own meet the access policy, users can decrypt the ciphertext. In a PHR system, the patient's medical records are encrypted by ABE, and the fine-grained access control to the encrypted medical records is realized. As the PHR system has high requirements for data privacy and security, and the patient's records stored in the system is of great value, there are often illegal users maliciously divulging the data in the system for their own interests. Specifically, there are two categories of key abuse problems in CP-ABE. (1) Driven by personal interests, authorized users may reveal their private keys to illegal users [6, 14]. (2) The semi-trusted authority may redistribute the private keys to unauthorized users for the same reason. In order to solve the problem of key abuse, accountable ABE is proposed. In addition, it is common for patients to exit from the system. So how to revoke the user efficiently is also a research hotspot in ABE. Revocable ABE arises at the historic moment. In this paper, we aim to implement an accountable and full secure ABE scheme in the personal health system, which can publicly audit traceable users and support the revocation of malicious users. The patient's personal health records are encrypted and placed on the cloud server. Patients make access policies to allow specific people to access their health records. If there is malicious behavior that the user leaks the patient's medical records, it can be traced according to the identity-related information contained in the ciphertext. In order to ensure the security of the system, the system access rights of malicious users are reclaimed by indirect revocation. The proposed scheme is based on the personal health medical record system, which greatly protects the privacy of patients and the security of the system.

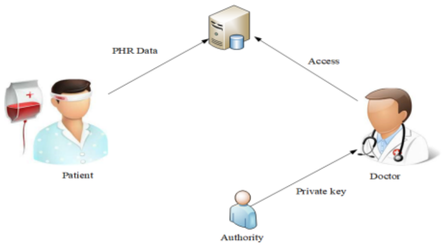


Fig. 1. PHR System

1.1 Related Work

ABE is evolved from fuzzy identity-based encryption [15]. After that, many ABE schemes were put forward to improve performance and security. In this chapter, we mainly talk about the relevant work to handle the matters of key abuse and user revocation.

1.1.1 Accountability

Due to the matter of key abuse in CP-ABE, accountability has become one of the criteria to measure the practicability of the scheme. The first key accountable ABE scheme was put forward in [16], whereas only supports the access policy expressed by "AND gate and wildcard". Constructing a traceable CP-ABE scheme is an adequate means to settle the matter of key abuse. There appeared two kinds of traceability: white-box traceability as well as black-box traceability. In 2013, black-box traceable CP-ABE [17] as well as white-box traceable CP-ABE [18] are put forward, and support monotonous access structure. A multi-authority CP-ABE syetem is brought forward by Li et al. [19]. Although the above two schemes are achieved under prime order groups, they only support the access structure of multi-valued "AND" gates with wildcards, and only achieve the selective security under the standard model, so the expression ability and security of access policies are relatively weak. The constructions in [17-21]only implements user traceability, but does not realize authority accountability. Ning et al. [22] proposed an attribute-based encryption scheme which supports both user traceability and authority accountability for the first time. The scheme in [23] allows the key generation center (KGC) and the attribute authority (AA) to jointly generate the user's private key, thus ensuring that KGC and AA cannot distribute the user's private key to unauthorized users. However, the scheme does not solve the problems of user key abuse and user revocation. Ning et al. [24] proposed a fully secure white-box traceable CP-ABE system for the first from non-interactive commitments. Zhao et al. [25] proposed a large universe CP-ABE with black-box traceability. In this scheme, the size of public parameters not to grow linearly with the number of attributes.

1.1.2 Revocation

In a CP-ABE system, there are often cases such as user privilege change, user exit and user private key disclosure, so it is indispensable to consider the matter of user revocation. In the system, revoked user cannot decrypt any ciphertext. Meanwhile, the system permissions of other unrevoked users in the system are not affected. In 2006, revocable attribute-based encryption (RABE) was put forward for the first time in [26]. Goyal et al. [27] put forward an ABE scheme that implements indirect attribute revocation. In their construction, each attribute has a time tag representing the validity period. When the system time exceeds the time available for an attribute, the attribute is revoked. In the construction in [28], a CP-ABE scheme for user revocation using binary tree is proposed, but its performance is not high, and greatly increases the calculation and communication burden of the key generation center. Liu et al. [29] implement user revocation by setting an effective access time to the root node of the access control tree, but the cost of key management and decryption is high. A directly RABE supporting verifiable ciphertext agents is proposed in the construction in [30]. However, if there are a lot of user attributes, direct revocation will increase the burden on the cryptographer because the system has to update the user revocation list, thus reducing the feature of the entire system. Indirect revocation is generally implemented by an authority or a third-party agent, and the computational burden is small.

However, until now, there is rarely a full secure ABE scheme that supports user traceability, authority accountability and user revocation concurrent that prevents the practical application of ABE in PHR system.

1.2 Our Construction

In this article, we concentrate on the two common matters of key abuse and user revocation in existing CP-ABE schemes and come up with a new white-box accountable ABE scheme with public auditing and user revocation.

Our scheme realizes both traceable users and accountable authority. It solves the problem of key abuse of untrusted authority, which is often ignored in the existing CP-ABE schemes. When tracing malicious users, we utilize Paillier-style encryption as an extractable commitment. Because there is no requirement to keep an identity table for traceability, there is no traceability storage, which greatly saves the storage space of the server. In addition, an auditor is used to determine whether the caught user is innocent or guilty. What's more, we adopt the revocation method based on the trust tree to realize the revocation of misbehaving users, authorities and other users who quit the system. At the same time, it is proved the scheme is full secure in the standard model.

Comparing the scheme with others on both theoretically and experimentally. Through the analysis, it can be concluded that our scheme has more complete functions and higher security. In terms of efficiency, although our scheme consumes more time than the schemes in [22, 31], it is acceptable because of the supplement function of the user revocation .

2. Preliminary

2.1 Linear Secret-sharing Scheme

If it meets these requirements as below, the secret sharing scheme Π over a series of parties E is regarded as linear.

(a) The shares for all parties constitute a phasor on Z_p.

(b) For Π, there is a $l \times n$ matrix A called the sharing-generating matrix. For all $i = 1, \ldots, l$, $\delta(i)$ (δ is a mapping from $\{1, \ldots, l\}$ to E) is the label for the i th row of A. We consider the column vector $v = (s, r_2, r_3, \ldots, r_n)^{\text{ú}}$, where $s \in Z_p$ is the shared secret and $r_2, r_3, \ldots, r_n$ are randomly chosen from Z_p. Then Av is the vector of l shares of the secret s on the basis of Π. The share $(Av)_i$ pertains to party $\delta(i)$.

2.2 Trusted Tree-based Revocation Approach

We use the subset-cover algorithm $KUNode(st, rl, t)$ [32, 33] to revoke a user, where st is the data structure of the tree, rl signifies a revocation entry with the identity of the revoked user while t signifies the most recent revocation period. The user will be distributed an identity id and an undefined leaf node when s/he joins the system. The leaf node is identified by id . The implementation of user revocation claims the user id to save secret keys in Path(id). The Path(id) represents all nodes id from the root node to the leaf node.

2.3 Composite Order Bilinear Groups

ψ represents a group generator. ψ inputs a security parameter ξ and outputs a group G of order $N = n_1 n_2 n_3$, where n_1, n_2, n_3 are different primes.

Let G, G_T are cyclic groups of order $N = n_1 n_2 n_3$, and e: $G \times G \to G_T$ is a bilinear mapping, which meets the characters as following:

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

(b) Bilinearity : $\forall x, y \in G$,$c, d \in Z_p$, the equation $e(x^c, y^d) = e(x, y)^{cd}$ is true [34].

(c) Computability: the map $e: G \times G \to G_T$ can be effectively calculated.

Let $G = G_{n_1} G_{n_2} G_{n_3}$, G_{n_1}, G_{n_2} and G_{n_3} are the subgroups of order n_1, n_2 and n_3 in G, severally. Suppose g is a generator of group G, while $g^{n_2 n_3}$ is the generator of subgroup G_{n_1} accordingly. Similarly, $g^{n_1 n_3}$ is the generator of subgroup G_{n_2}, and $g^{n_1 n_2}$ is the generator of subgroup G_{n_3}. Therefore, there exists $\alpha_1, \alpha_2 \in Z_N$, such that $f_1 = (g^{n_1 n_2})^{\alpha_1}$, $f_2 = (g^{n_1 n_2})^{\alpha_2}$. At this time, there are: $e(f_1, f_2) = e(g^{\alpha_1}, g^{n_3 \alpha_2})^{n_1 n_2 n_3} = 1$.

If we choose $i,, j$, $f_i \in G_{n_i}$, $f_j \in G_{n_j}$, then $e(f_i, f_j)$ is the unit component in the group G_T. It also shows that the composite order subgroups G_{n_1}, G_{n_2} and G_{n_3} are orthogonal to each other.

2.4 Complexity Assumptions

Assumption 1 ($\mathit{Subgroup\ Decision\ Problem\ for}\ 3\ \mathit{primes}$). Provided a group generator ψ , we distribute these parameters as follows [35]:

$$\mathcal{G} = (G, G_T, N = n_1 n_2 n_3, e) \xleftarrow{r} \psi,\ g \xleftarrow{r} G_{n_1}, W_3 \xleftarrow{r} G_{n_3},$$
$$B = (\mathcal{G}, g, W_3),\ E_1 \xleftarrow{r} G_{n_1, n_2},\ E_2 \xleftarrow{r} G_{n_1}.$$

$\mathcal{A}$ has the following advantages in breaking this assumption. It is defined:

$$Adv1_{\psi,\mathcal{A}}(\xi) = | Pr[\mathcal{A}(B, E_1) = 1] - Pr[\mathcal{A}(B, E_2) = 1] | .$$

Definition 1. If for any PPT algorithm $\mathcal{A}$, $Adv1_{\psi,\mathcal{A}}(\xi)$ is negligible of ξ, we call ψ meets Assumption 1.

Assumption 2. Provided a group generator ψ, we distribute these parameters as below [35]:

$$\mathcal{G} = (G, G_T, N = n_1 n_2 n_3, e) \xleftarrow{r} \psi,\ g, W_1 \xleftarrow{r} G_{n_1}, W_2, V_2 \xleftarrow{r} G_{n_2}, W_3, V_3 \xleftarrow{r} G_{n_3},$$
$$B = (\mathcal{G}, g, W_1 W_2, W_3, V_2 V_3),\ E_1 \xleftarrow{r} G,\ E_2 \xleftarrow{r} G_{n_1, n_3}.$$

$\mathbb{A}$ has the following advantages in breaking this assumption. It is defined:

$$Adv2_{\psi,\mathbb{A}}(\xi) = |Pr[\mathbb{A}(B, E_1) = 1] - Pr[\mathbb{A}(B, E_2) = 1]|.$$

Definition 2. If for any PPT algorithm $\mathbb{A}$, $Adv2_{\psi,\mathbb{A}}(\xi)$ is negligible of ξ, we call ψ meets Assumption 2 .

Assumption 3. Provided a group generator ψ, we distribute these parameters as follows [35]:

$$\mathbb{G} = (G, G_T, N = n_1n_2n_3, e) \xleftarrow{r} \psi,\ \alpha, s \xleftarrow{r} Z_N,$$
$$g \xleftarrow{r} G_{n_1}, W_2, V_2, X_2 \xleftarrow{r} G_{n_2}, W_3 \xleftarrow{r} G_{n_3},$$
$$B = (\mathbb{G}, g, g^{\alpha}W_2, W_3, g^{s}V_2, X_2),\ E_1 = e(g,g)^{\alpha s},\ E_2 \xleftarrow{r} G_T.$$

$\mathbb{A}$ has the following advantages in breaking this assumption. It is defined:

$$Adv3_{\psi,\mathbb{A}}(\xi) = |Pr[\mathbb{A}(B, E_1) = 1] - Pr[\mathbb{A}(B, E_2) = 1]|.$$

Definition 3. If for any PPT algorithm $\mathbb{A}$, $Adv3_{\psi,\mathbb{A}}(\xi)$ is negligible of ξ, we call that ψ meets Assumption 3.

3. System model and security model

3.1 Entities in the System

This system includes five entities, namely, authority, PHR server, data owner, data user and auditor as represented in **Fig. 2**. Their functions and responsibilities are as follows:

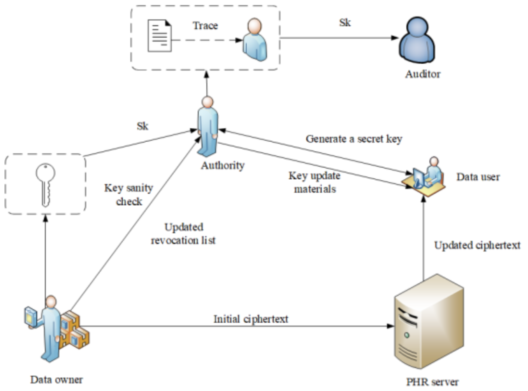


Fig. 2. System Model

Authority. In this system, the authority (AT) is considered not to be trusted. AT produces public parameters pp and interacts with a data user (DU) to produce secret key sk_{id}. In addition, when user revocation occurs, AT is responsible for broadcasting the key update material.

PHR server. PHR server stores the beginning ciphertext BT . Moreover, the PHR server updates BT to the latest ciphertext at the current time.

Data owner. Plaintext is encrypted by the data owner (DO) to BT and sends it to the PHR server for storage. Furthermore, in order to implement user revocation, DO renews revocation list rl and sends it to AT . To determine whether the compromised secret key is intact, DO performs the key sanity check.

Data user. In this system, unrevoked users and revoked users are two genres of data users. If a user is unrevoked and his attributes meet the access structure, he could access. Unrevoked users updates his own decryption key and decrypt the ciphertext through the key update material broadcast by AT .

Auditor. The auditor acts as a fair adjudicator in the system. When the traced user denies disclosing the private key, the auditor executes the audit algorithm to determine whether the user has been framed or innocent.

3.2 System Model

We will elaborate specifically on our scheme with traceable user, accountable authority, public auditing, user revocation and no storage for tracing in this section. Our scheme contains a total of ten algorithms as below:

$Setup(\xi,\tau,V,U,)\rightarrow(pp,st,rl,msk)$. AT launches algorithm $Setup$ and inputs security parameter ξ, system lifetime τ , the universe of attributes V as well as the amount of system users U , and outputs public parameters pp , state st , and master secret key msk . Besides, it constitutes a revocation list $rl=\varnothing$.

$KeyGen(pp,st,msk,S,id)\rightarrow(sk_{id})$. It is executed by DU and AT together. It takes public parameters pp , state st , master secret key msk , an attribute set S , the identifier of DU id $(id\in Z_N^*)$ as input, and outputs a secret key sk_{id} .

$Key\ update(I,rl,t,(id_i,t_i),st,msk)\rightarrow(kd_t,rl)$. This algorithm includes two sub-algorithms: $Rev(rl,(id_i,t_i))\rightarrow rl$ and $KeyUpdate(st,rl,msk,t)\rightarrow kd_t$. The algorithm inputs a series of identifiers I , rl , the current revocation time t, revocation epoch (id_i,t_i) , state st , msk , and outputs a key updating ingredient kd_t as well as the updated revocation list rl corresponding to t .

$Decryption\ key\ generation(pp,sk_{id},kd_t)\rightarrow dk_{id,t}/\perp$. DU is responsible for executing the algorithm. It takes pp , sk_{id} , the key updating kd_t as input. If DU is unrevoked user in this period t and S meets the access policy, it outputs decryption key $dk_{id,t}$. Otherwise, it outputs the failure symbol $\perp$.

$Encrypt((A,\rho),pp,t,m,msk)\rightarrow BT$. This algorithm takes access structure (A,ρ) , public parameters pp , current time t , plaintext m , msk as input, and the beginning ciphertext BT as output.

$Ciphertext\ update(BT,pp,t')\rightarrow CT/\perp$. Algorithm $Ciphertext\ update$ is carried out by PHR server. The algorithm inputs the latest ciphertext BT , the public parameters pp as well as the recent revocation epoch $t'\in\tau$. The updated ciphertext CT or $\perp$ is taken as output.

$Decrypt(CT, pp, dk_{id}) \to m/\perp$. DU implements this algorithm. It takes the latest ciphertext CT, public parameters pp, decryption key dk_{id} as input and plaintext m or $\perp$ as output.

$Key\ sanity\ check(sk_{id}, pp) \to 1/0$. This algorithm launched by AT. And it is utilized to ensure that the key sk_{id} is in well form during the decryption course. The algorithm inputs sk_{id}, public parameters pp. If sk_{id} fails this check, it outputs 0. If not, the output is 1.

$Trace(sk_{id}, pp, msk) \to id/\perp$. AT implements algorithm $Trace$. It inputs the secret key sk_{id}, pp, msk. If algorithm $Key\ sanity\ check$ outputs 0, it indicates sk_{id} is not well-formed. It is not necessary for the $Trace$ algorithm to continue to execute. The algorithm outputs $\perp$. On the contrary, if the output is 1, it indicates sk_{id} is well-formed. $Trace$ algorithm is executed to extract and output the user's id from sk_{id}.

$Audit(pp, sk_{id}, sk_{id}^{*}) \to guilty/acquitted$. The algorithm is implemented by DU and the auditor together. If a user is caught by $Trace$ algorithm, but he does not admit his crime. At this time, the $Audit$ algorithm is used to determine whether the user has been wronged or guilty.

3.3 Security Model

To manifest the security of proposed scheme, a security game, namely, the IND-CPA game is defined. The specific description is as below:

The IND-CPA game. This is an indistinguishability under chosen-plaintext attack game. It's a standard semantic security concept that any CP-ABE scheme must meet.

Setup. The challenger $\mathbb{C}$ executes algorithm $Setup$, holds msk in private and releases pp to the adversary $\mathbb{A}$.

Query 1. K key query requests with a series of attributes $(id_1, S_1), (id_2, S_2)\ldots(id_k, S_k)$ are send to $\mathbb{C}$ by $\mathbb{A}$. After obtaining these requests, $\mathbb{C}$ carries out algorithm $KeyGen$, $Key\ update$, $Decryption\ \ key\ generation$ and delivers decryption key to $\mathbb{A}$.

Challenge. $\mathbb{A}$ sends an access structure which isn't subject to the above k attribute sets and two messages of the same length m_0, m_1 to $\mathbb{C}$. After randomly tossing a coin $\lambda \in \{0,1\}$, $\mathbb{C}$ encrypts m_λ. The ciphertext is transmitted to $\mathbb{A}$.

Query 2. Identical with Query 1.

Guess. $\mathbb{A}$ makes a guess λ' that λ is 0 or 1.

In this process, $Adv = |\Pr[\lambda' = \lambda]\text{-}\frac{1}{2}|$ is taken as the advantage of $\mathbb{A}$. We allege our scheme is full secure if $\mathbb{A}$ has a at the utmost negligible advantage can win the game in any PPT.

4. Construction

In this section, we propose an accountable attribute-based encryption scheme in the personal health record system. The proposed scheme realizes public auditing and user revocation. In addition, there is no storage for retro in the scenario. The details of the proposed scheme are as follows.

4.1 $\mathbf{Setup}(\xi, \tau, \mathbf{V}, \mathbf{U},) \to (\mathbf{pp}, \mathbf{st}, \mathbf{rl}, \mathbf{msk})$

Get a bilinear group map $\mathbb{G} = \{e, G, G_{n_i}, N, n_i\}$, where g, g_3 are the generator of $G_{n_1}, G_{n_3}, N = n_1 n_2 n_3, n_i$ are the order of group G, G_{n_i} severally.

Then, the algorithm selects randomly $\alpha, \beta, \gamma, u, \eta \in Z_N^*, v, u_0, \ldots, u_D$ (D denotes the size of $\tau) \in G_{n_1}$ and chooses $v_i \in Z_N^*$ randomly for every attribute $i \in V$.

Select randomly b, d two prime numbers $, \gcd(bd, (d-1)(b-1)) = 1$ and $|b| = |d|, b \neq d$. Let $\pi = lcm(d-1, b-1), n = bd, Q = \pi^{-1}$ mod n and $g_1 = (n+1)$.

It selects TC binary tree with more than U leaves. Finally, it returns public parameters $pp = (N, n, g_1, v, u_0, \ldots, u_D, g, g^{\beta}, g^{\gamma}, g^{u}, e(g,g)^{\alpha}, e(g,g)^{\eta}, \{V_i = g^{v_i}\}_{i \in V})$, TC as state st, revocation list $rl = \varnothing$ and $msk = (p, q, \eta, \alpha, g_3)$ as master secret key.

4.2 $\mathbf{KeyGen}(\mathbf{pp}, \mathbf{st}, \mathbf{msk}, \mathbf{S}, \mathbf{id}) \to (\mathbf{sk_{id}})$

First, the algorithm selects randomly an unallocated leaf node from TC. This node is utilized to keep id. The algorithm operates as below for every node θ in Path(id).

It retrieves η_θ from the node θ. It randomly chooses and saves $\eta_\theta \in Z_N^*$ in the node θ if η_θ is unavailable.

A user DU identified by id and the authority AT interact to produce the key as the following steps. To make the whole process clearer, the flow of the domain key generation phase is shown in **Fig. 3**.

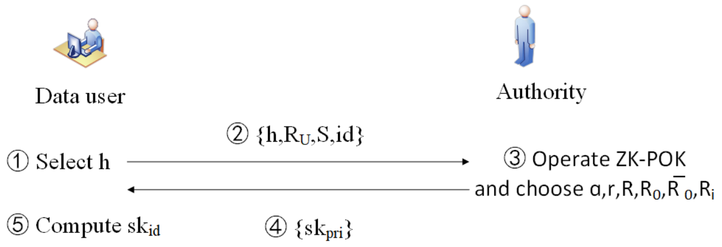


Fig. 3. The Flow of the Domain Key Generation Phase

For DU:

DU selects randomly $h \in Z_N^*$ and computes $R_U = g^h$.

Then, g^h, a set of attributes S as well as id are sent to AT.

Next, AT operates a ZK-POK of the discrete log of R_U in relation to g.

For AT **:**

AT examines if the ZK-POK is available or not. If the examination succeeds, it executes the next step. If not, AT discontinues the interaction.

Then, it chooses randomly $a \in Z_N^*, r \in Z_N^*$ and $R, R_0, \overline{R_0}, \{R_i\}_{i \in S} \in G_{n_3}$.

Next, the primary secret key sk_{pri} for each user with id and S is computed as

$$\{S, \overline{K} = g^{\frac{\alpha}{\beta+\overline{T}}}(g^h)^{\frac{\gamma}{\beta+\overline{T}}} v^a R, \overline{T} = g_1^{id} r^n \ mod\ n^2, \overline{L} = g^a R_0, \overline{L_1} = g^{\beta a} \overline{R_0},$$
$$\{\overline{K_i} = V_i^{(\beta+\overline{T})a} R_i\}_{i \in S}, \{g^{\eta_\theta}\}_{\theta \in Path(id)}\}$$

Finally, AT sends (a, sk_{pri}) to DU .

For DU **:**

DU determines whether the following equations are true.

(a) $e(\overline{L_1}, g) = e(g^\beta, \overline{L}) = e(g^\beta, (g)^a)$.

(b) $e(\overline{K}, g^\beta g^{\overline{T}}) = e(\overline{L_1}(\overline{L})^{\overline{T}}, v) e(R_U, g^\gamma) e(g, g)^\alpha$.

(c) $s.t.\ e(V_x, \overline{L_1}(\overline{L})^{\overline{T}}) = e(\overline{K}_x, g), \ \exists x \in S$.

If all the equations hold, DU holds on the interaction and computes $h_{id} = \frac{a}{h}$. Then, it takes his secret key sk_{id} as below:

$$sk_{id} = \{S, K = \overline{K}(g^u)^{h_{id}}, T = \overline{T}, L = \overline{L}, L_1 = \overline{L_1}, R_U, h_{id}, \{K_i = \overline{K_i}\}_{i \in S}, \{g^{\eta_\theta}\}_{\theta \in Path(id)}\} .$$

Otherwise, DU aborts the interaction.

4.3 Key update$(\mathbf{I}, \mathbf{rl}, \mathbf{t}, (\mathbf{id_i}, \mathbf{t_i}), \mathbf{st}, \mathbf{msk}) \rightarrow (\mathbf{kd_t}, \mathbf{rl})$

Firstly, AT operates the revocation algorithm for every identifier and revocation epoch (id_i, t_i) . To make the whole process clearer, the flow of the domain key generation phase is shown in **Fig. 4**.

$Rev(rl, (id_i, t_i)) \rightarrow rl$. This algorithm is mainly used to update revocation lists in systems. DO launches the revocation algorithm and inputs the revocation list rl , the revocation epochs (id_i, t_i) and outputs the renewed rl as follows: $rl \cup (id_i, t_i) \rightarrow rl$. **Fig. 5** shows the composition and update process of the revocation list. After obtaining the latest revocation list rl , AT executes the $KeyUpdate$ algorithm.

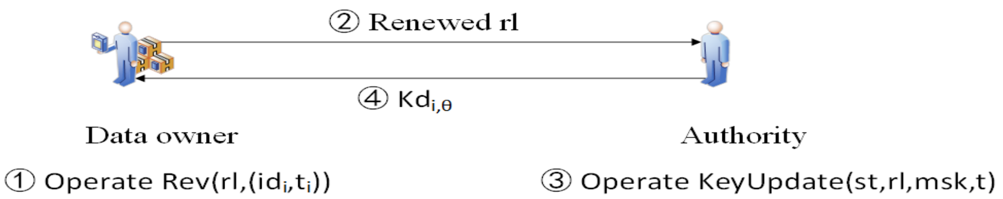
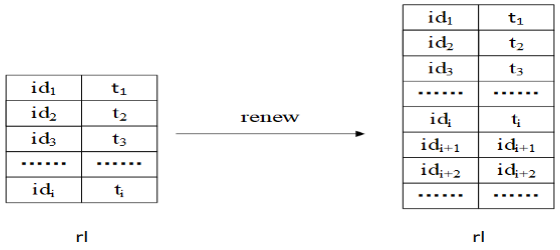

② Renewed rl
④ Kd$_{i,\theta}$
Data owner
Authority
① Operate Rev(rl,(id$_i$,t$_i$))
③ Operate KeyUpdate(st,rl,msk,t)

Fig. 4. The Flow of the Domain Key Update Phase



id₁	t₁
id₂	t₂
id₃	t₃
......	......
idᵢ	tᵢ

id₁	t₁
id₂	t₂
id₃	t₃
......	......
idᵢ	tᵢ
idᵢ₊₁	idᵢ₊₁
idᵢ₊₂	idᵢ₊₂
......	......

Fig. 5. Renew

$KeyUpdate(st, rl, msk, t) \rightarrow kd_t$: The algorithm inputs state st , the latest revocation list rl , master secret key msk , time $t \in \tau$, and outputs the key updating material $kd_{t,\theta}$. In this algorithm, firstly, the time coding method proposed in reference [36] is used to encode the time. The time t is encoded as a bit $\tilde{t}$. Let $\zeta \in [D]$ be the set of all indexes i meeting $t[i]$ = 0. The key updating material kd_t is generated as follows for every node $\theta \in KUNode(st, rl, t)$: retry η_θ . Randomly select $\mu \in Z_N^*$ and output the key updating $kd_{t,\theta}$ as below: $kd_{t,\theta} =< kd_1, kd_2 >=< g^{\eta-\eta_\theta}(u_0\Pi_{i\in\zeta}u_i)^{\mu}, g^{\mu} >$.

4.4 Decryption key generation$(\mathbf{pp}, \mathbf{sk_{id}}, \mathbf{kd_t}) \rightarrow \mathbf{dk_{id,t}} / \perp$

Let X and Y represent the sets Path(id) and $KUNode(st, rl, t)$, separately. If X ∩ Y =∅, the algorithm returns a failure symbol $\perp$. If not, we can obtain node $\theta \in X$ ∩ Y.

Randomly select $\mu' \in Z_N^*$ and calculate the decryption key $dk_{id,t}$ as follows:

$$D_1 = g^{\eta_\theta} \cdot kd_1 \cdot (u_0\Pi_{i\in\zeta}u_i)^{\mu'} = g^{\eta_\theta} \cdot g^{\eta-\eta_\theta}(u_0\Pi_{i\in\zeta}u_i)^{\mu} \cdot (u_0\Pi_{i\in\zeta}u_i)^{\mu'} = g^{\eta}(u_0\Pi_{i\in\zeta}u_i)^{\mu+\mu'}$$

$$D_2 = kd_2 \cdot g^{\mu'} = g^{\mu} \cdot g^{\mu'} = g^{\mu+\mu'}$$

Finally, the algorithm outputs: $dk_{id,t} = (S, K, T, L, L_1, R_U, h_{id}, \{K_i\}_{i\in S}, g^{\eta_\theta}, D_1, D_2)$.

4.5 Encrypt$((\mathbf{A}, \rho), \mathbf{pp}, \mathbf{t}, \mathbf{m}, \mathbf{msk}) \rightarrow \mathbf{BT}$

For the ciphertext related to attributes, the algorithm chooses randomly $\vec{y} = (s, y_2, y_3, \ldots, y_n)^{\mathrm{T}}$, $r_j \in Z_N^*$ for each row A_j of A, where s is randomly selected as a secret value. The attribute-related ciphertext is calculated as:

$$(C_0 = g^s, C_1 = (g^{\beta})^s, C_2 = (g^{\gamma})^s, C_3 = (g^{u})^s, C = m \cdot e(g,g)^{\alpha s} \cdot e(g,g)^{\eta s},$$
$$\{C_{j,1} = v^{A_j\vec{y}} V_{\rho(j)}^{\ -r_j}, C_{j,2} = g^{r_j}\}_{j\in[l]})$$

For the ciphertext related to time, the algorithm encodes t to the bit representation $\tilde{t}$. Then it derives the time $CTEncode(\tilde{t}, \tau)$ [36] $\rightarrow \tilde{t}$ and let $\zeta \in [D]$ be the set of each index i' satisfying $t[i']$ = 0. Then, it calculates the ciphertext affiliated to time as follow:

$C_4 = u_0^s, C_{5,i} = u_i^{\ s}, i \in \zeta$.

Finally, it outputs the ciphertext: $CT = (C_0, C_1, C_2, C_3, C, C_4, \{C_{5,i}\}_{i\in\zeta}, \{C_{j,1}, C_{j,2}\}_{j\in l})$.

4.6 $\mathbf{Ciphertext\ update}(\mathbf{BT}, \mathbf{pp}, \mathbf{t}') \to \mathbf{CT}/\perp$

The output of this algorithm is in the following two cases. If the ciphertext is invalid or the timestamp in the latest ciphertext CT is bigger than the time t', the algorithm will discontinue and output $\perp$. If not, it will encode t' to $\tilde{t}'$ and update ciphertext CT .

Let $\zeta \in [D]$ signify the set of each index i' satisfying the condition $t[i']$ = 0.

Next, the PHR server calculates the ciphertext related to time as below:

$$C_t = C_4 \Pi_{i\in\zeta} C_{5,i} = (u_0 \Pi_{i\in\zeta} u_i)^s$$

Next, the algorithm chooses randomly $\vec{y}' = (s', y_2^{'}, y_3^{'}, \ldots, y_n^{'})^{\mathsf{T}} \in Z_N^*$.

Then, it calculates the ciphertext:

$$C_{0'} = C_0 \cdot g^{s'} = g^{s+s'} \qquad C_{t'} = C_t \cdot (u_0 \Pi_{i\in\zeta} u_i)^{s'} = (u_0 \Pi_{i\in\zeta} u_i)^{s+s'}$$

Finally, it outputs:

$$CT = (C, C_{0'}, C_1, C_2, C_3, C_{j,1}, C_{j,2}, C_{t'}).$$

4.7 $\mathbf{Decrypt}(\mathbf{CT}, \mathbf{pp}, \mathbf{dk_{id}}) \to \mathbf{m}/\perp$

After DU obtains the latest ciphertext from the PHR server, it inputs pp , dk_{id} and the latest ciphertext CT . Then, if the time t in $dk_{id,t}$ doesn't match the time t in CT or S doesn't satisfied (A, ρ), it output is $\perp$. If not, this algorithm will output plaintext message m through the following operations.

First, it calculates the constants $\omega_j \in Z_N^*$ that satisfies $\Sigma_{\rho(j)\in S}\omega_j A_j = (1, 0, \ldots, 0)$ and the part related to the attributes:

$$H_a = e((C_0)^T C_1, K)(e(C_2, R_U)e(C_3, (g^T g^\beta)^{h_{id}}))^{-1}$$

$$H_b = \Pi_{\rho(j)\in S}(e(C_{j,1}, L^T L_1)e(K_{\rho(j)}, C_{j,2}))^{\omega_j}$$

$$H = \frac{H_a}{H_b} = \frac{e(g,g)^{s\alpha}e(g,v)^{(T+\beta)sa}}{e(v,g)^{sa(T+\beta)}} = e(g,g)^{sa}$$

Next, it acquires the hiding component in plaintext:

$$E = \frac{e(D_1, C_0)}{e(D_2, C_t)} = e(g,g)^{s\eta} \qquad m = \frac{C}{H \cdot E}$$

Finally, it outputs the plaintext m .

Correctness

H_a=$e(g^{sT}g^{\beta s}, g^{\frac{\alpha}{T+\beta}}(g^h)^{\frac{\gamma}{\beta+T}}v^a R(g^u)^{h_i d})(e(g^{\gamma s}, g^h)e(g^{us}, (g^T g^\beta)^{h_{id}}))^{-1}$

$=e(g,g)^{s(T+\beta)uh_{id}}e(g,g)^{s\alpha}e(g,g)^{s\gamma h}e(g,v)^{(T+\beta)sa}(e(g,g)^{h\gamma s}e(g,g)^{ush_{id}(T+\beta)})^{-1}$

$=e(g,g)^{s\alpha}e(g,v)^{(T+\beta)sa}$

H_b=$\prod_{\rho(j)\in S}(e(v^{A_j\vec{y}}V_{\rho(j)}{}^{-r_j}, (g^a R_0)^T g^{\beta a}\bar{R}_0)e(V_{\rho(j)}{}^{(\beta+\bar{T})a}R_i, g^{r_j}))^{\omega_j}$

$=\prod_{\rho(j)\in S}(e(v^{A_j\vec{y}}V_{\rho(j)}{}^{-r_j}, g^{aT}g^{\beta a})e(V_{\rho(j)}{}^{(\beta+\bar{T})a}, g^{r_j}))^{\omega_j}$

$=\prod_{\rho(j)\in S}(e(v^{A_j\vec{y}}, g^{a(T+\beta)}))^{\omega_j}$

$=e(v,g)^{a(T+\beta)\sum_{\rho_j\in S}A_j\vec{y}\omega_j}$

$=e(v,g)^{sa(T+\beta)}$

$$E=\frac{e(D_1,C_0)}{e(D_2,C_t)}=\frac{e(g^\eta(u_0\Pi_{i\in\zeta}u_i)^{\mu+\mu'},g^s)}{e(g^{\mu+\mu'},(u_0\Pi_{i\in\zeta}u_i)^s)}=\frac{e(g^s,(u_0\Pi_{i\in\zeta}u_i)^{\mu+\mu'})e(g^s,g^\eta)}{e(g^{\mu+\mu'},(u_0\Pi_{i\in\zeta}u_i)^s)}=e(g,g)^{s\eta}$$

4.8 Key sanity check$(\mathbf{sk_{id}}, \mathbf{pp}) \rightarrow \mathbf{1/0}$

The key sanity check of sk_{id} includes the following four phases.

Firstly, check whether sk_{id} is structure of $(S,K,T,L,L_1,R_U,h_{id},\{K_i\},g^{\eta_\theta})$ and $T\in Z_N^*,K,L,L_1,R_U,\{K_i\}_{i\in S}\in G,g^{\eta_\theta}\in G_{n_1}$.

- $e(g,L_1)=e(g^\beta,L)$.

- $e(g^\beta g^T,K)=e((g^\beta g^T)^{h_{id}},g^u)e(R_U,g^\gamma)e(L_1L^T,v)e(g,g)^\alpha$.

- $\exists x\in S, s.t.\, e(V_x,L_1L^T)=e(K_x,g)$.

If sk_{id} passes this check, the output is 1. If not, the output is 0.

4.9 Trace$(\mathbf{sk_{id}}, \mathbf{pp}, \mathbf{msk}) \rightarrow \mathbf{id}/\perp$

If algorithm *Key sanity check* output is 0, it implies that sk_{id} is not well-formed and doesn't deserve to trace. Hence, the algorithm outputs $\perp$. If not, AT will perform the operations $Q=\pi^{-1}\ mod\ n, T=g_1^{id}r^n\ mod\ n^2$ to extract and output the identity id . It can obtain the result from above two equation:

$$T^{\pi Q}=g_1{}^{id\cdot\pi Q}\cdot r^{n\cdot\pi Q}=g_1{}^{id}=1+id\cdot n\ mod\ n^2; id=\frac{((T)^{\pi Q}\ mod\ n^2)-1}{n}\ mod\ n.$$

4.10 Audit$(\mathbf{pp}, \mathbf{sk_{id}}, \mathbf{sk_{id}^*}) \rightarrow \mathbf{guilty/acquitted}$

DU is considered as a malicious user, but it declares to be acquitted or defamed. In this case, an auditor is needed to determine if the user is guilty or not. We propose an auditing algorithm that anyone can act publicly as a auditor in the system. After the *Trace* algorithm outputs the traced key sk_{id}^*, the auditor will interact with DU as follows.

First, DU releases to his secret key sk_{id} to the auditor. If algorithm $Key\ sanity\ check$ outputs 0, the auditor discontinues. Otherwise, the auditor proceeds to the next step.

The auditor checks whether h_{id} is equal to h_{id}^{*}. If equal, it means that the user does reveal the secret key sk_{id} to others. Consequently, DU is guilty. The output is $guilty$. If not, the output is $acquitted$.

5. Security Analysis

5.1 IND-CPA Security

In this part, the security proof of proposed scheme will be put forward. The security of our new accountable authority, traceable user and revocable user CP-ABE scheme (referred to as **AATR-ABE**) is based on IND-CPA security of the ABE scheme [35] (referred to as Lewko ABE).

Lemma 1. [35] Lewko ABE is secure if Assumption 1, 2 and 3 in Subsection **2.5** are true.

Lemma 2. [35] **AATR-ABE** is secure in the IND-CPA game of Subsection **3.3** if Lewko ABE is secure.

Theorem 2. AATR-ABE is secure if Assumption 1, 2 and 3 in Subsection 2.5 are true.

Proof. After exploiting A who has a non-negligible advantage to win the IND-CPA game of **AATR-ABE**, we establish a PPT simulator algorithm T to break Lewko ABE.

Setup: Lewko ABE gives public parameters $pp=(g,e(g,g)^{\alpha},g^{\kappa},N,\{V_i=g^{v_i}\}_{i\in V})$ to T, T chooses α,γ at randomly from Z_N^* and two random primes b,d which meets $|b|=|d|,b\neq d$, $\gcd(bd,(d-1)(b-1))=1$. Let $\pi=lcm(d-1,\ b-1),n=db$, $Q=\pi^{-1} \bmod n$ and $g_1=(n+1)$. T sends $pp=(N,n,g_1,v=g^{\kappa},u_0,\ldots,u_D,g,g^{\beta}, g^{\gamma},g^{u},e(g,g)^{\alpha},e(g,g)^{\eta},\{V_i=g^{v_i}\}_{i\in V})$ to A .

Query 1: To query a decryption key, A sends (id,S) to T . T sends S to Lewko ABE. After receiving the S from T , Lewko ABE gives decryption key as $\mathrm{d}k=\{\tilde{K}=g^{\kappa\tilde{a}}g^{\alpha}R,\ \tilde{L}=g^{\tilde{a}}R_0,\{K_i=V_i^{\tilde{a}}R_i\}_{i\in S}\}$ to T . In Lewko ABE, the authority independently selects and distributes decryption keys to users. On the contrary, in **AATR-ABE**, key generation is caused by the interaction between the authority and an user. The secret key is affected by both h produced by the user and a produced by the authority. During key generation, first, the user randomly selects h and submits $R_U=g^{h}$ to the authority. The user runs a zero-knowledge proof about $<R_U,h>$, which indicates the presence of a knowledge extractor E. The authority could retrieve discrete logarithm h by using E . Accordingly, in IND-CPA game, T can retrieve h from R_U . T randomly selects $r\in Z_N^*$ and calculates $T=\bar{T}=g_1^{id}r^{n}\ mod\ n^2$ and $\dfrac{1}{\beta+T}\ mod\ N$. Let $c=\dfrac{\tilde{c}}{\beta+T}$, $h_{id}=\dfrac{a}{h}$. Then T randomly selects $\overline{R_0}\in G_{n_3}$ and calculates

$\bar{K}=(\tilde{K})^{\frac{1}{\beta+T}}(g^{h})^{\frac{\gamma}{\beta+T}}=g^{\frac{\alpha}{\beta+T}}v^{a}g^{\frac{\gamma h}{\beta+T}}R^{\frac{1}{\beta+T}},K=\bar{K}(g^{u})^{h_{id}}$, $\bar{L}=(\tilde{L})^{\frac{1}{\beta+T}}$ $=g^{a}(R_0)^{\frac{1}{\beta+T}},L=\bar{L}$, $\overline{L_1}=(\tilde{L})^{\frac{\beta}{\beta+T}}=g^{\beta a}R_0^{\frac{\beta}{\beta+T}}\overline{R_0},L_1=\overline{L_1},\{\overline{K_i}\quad=K_i=V_i^{(\beta+T)a}R_i\}_{i\in S}$, $\{K_i=K_i\}_{i\in S}$. T retrieves η_θ from the node θ. It randomly chooses and saves $\eta_\theta\in Z_N^*$ in the node θ if η_θ is unavailable. Finally, T gives A secret key $sk_{id,S}=(S,K,T,L,L_1,R_U,h_{id},\{K_i\}_{i\in S},g^{\eta_\theta})$. Retry η_θ for every node $\theta\in KUNode(st,rl,t)$, and randomly select $\mu\in Z_N^*$ and calculates $kd_{t,\theta}=<kd_1,\ kd_2>\ =<g^{\eta-\eta_\theta}(u_0\Pi_{i\in\zeta}u_i)^{\mu},g^{\mu}>$. Then, randomly select $\mu'\in Z_N^*$ and calculate the decryption key $dk_{id,t}$ as follows:

$$D_1=g^{\eta_\theta}\cdot kd_1\cdot(u_0\Pi_{i\in\zeta}u_i)^{\mu'}=g^{\eta}(u_0\Pi_{i\in\zeta}u_i)^{\mu+\mu'}\ D_2=kd_2\cdot g^{\mu'}=g^{\mu}\cdot g^{\mu'}=g^{\mu+\mu'}$$

Finally, T sends the decryption key $dk_{id,t}=(S,K,T,L,L_1,R_U,h_{id},\ \{K_i\}_{i\in S},g^{\eta_\theta}\ ,D_1\ ,D_2)$ to A.

Challenge: (A,ρ) and two messages m_0,m_1 of the same length are sent to T by A.

T sends m_0,m_1 and (A,ρ) to Lewko ABE. Then T gains the challenge ciphertext ct as below:

$$\{\tilde{C}=m_\lambda\cdot e(g,g)^{\alpha s},C_0=g^{s},\{C_{j,1}=g^{\kappa A_j\vec{y}}V_{\rho(j)}{}^{-r_j},C_{j,2}=g^{r_j}\}_{j\in[l]},(A^*,\rho)\}$$

T makes

$$C=\tilde{C}\cdot e(g,g)^{\eta s},C_0=C_0,C_1=(C_0)^{\beta}=g^{\beta s},C_2=(C_0)^{\gamma}=g^{\gamma s},C_3=(C_0)^{u}=g^{us},C_4=u_0^{s},$$

$$C_{5,i}=u_i{}^{s},\ C_{j,1}=C_{j,1}=v^{A_j\vec{y}}V_{\rho(j)}{}^{-r_j},C_{j,2}=C_{j,2}.$$

The challenge ciphertext is send to A by T.

$$ct=\{C,C_0,C_1,C_2,C_3,C_4,C_{5,i},\{C_{j,1},C_{j,2}\}\}$$

Query Phase 2: A and T continue the above queries and interactions.

Guess: A generates a bit λ' as the guessing of λ and send it to T. Then, T gives λ' to Lewko ABE. Because the assignment of public parameters, decryption key and challenge ciphertext in the above process is the same as in the real system, we can conclude that the advantage of A breaking **AATR-ABE** is equivalent to the advantage of A breaking Lewko ABE.

6. Comparison

Next, we will contrast our scheme with other relevant works [22, 24, 25, 31, 36, 37] from the aspects of functionality and efficiency.

6.1 Functionality

In **Table 1**, we contrast our scheme with the solutions implemented by [22, 24, 25, 31, 36, 37] . [22] realizes traceable user, accountable authority, public auditing and no storage for tracing, but it does not implement user revocation. Although [37] supports both user traceability and user revocation, it does not support accountable users and public auditing. The scheme is selective security under the standard model. The user revocation is achieved in [36] using an indirect revocation, but it can't solve the matter of key abuse. The scheme in [31] only implements traceable user and the storage overhead for traceability is linear. As mentioned earlier, a personal health record system with traceable user, accountable authority, public auditing, user revocation needs to be proposed. Obviously, only our scheme can achieve the above functionalities at the same time. [24] and [25] only implement traceable users.

6.2 Efficiency

The cost of other operations is very small compared with exponentiation operation as well as pairing operation, so we only consider exponential operation and pairing operation when comparing efficiency. We compared our scheme with others [22, 24, 25, 31, 36, 37]with respect to efficiency. The storage and transmission overhead comparison results are given in **Table 2**, including key length, ciphertext length. The computational complexity comparison results are given in **Table 3**, including the user-side overhead and authority center overhead in the key generation phase, as well as encryption and decryption costs.

Let L_{Z_P}, L_G, L_{G_T} intend the length of a component in group Z_N^*, G and G_T separately. $|U|$ represents the quantity of users in path (id). Let $|S|$ represent the size of the user's attribute set. l represents the quantity of rows in A. n represents the size of user attribute sets. $|D|$ represents the size of τ. During decryption and encryption, time spent on a pairing operation is expressed as P. The time overhead executing a pair operation in both G_T and G is represented as E_{G_T} and E_G respectively. From **Table 2** and **Table 3**, we can see that compared with several other schemes, our plan has some advantages in terms of key length, ciphertext length, encryption complexity and decryption complexity. This means our scheme can achieve relatively high efficiency.

Table 1. Function Comparison

Scheme	**Traceable user**	**Accountable authority**	**Storage for tracing**	**Auditing**	**Revocation**	**Security**
[36]	✗	✗	✗	✗	✓	selectively secure
[37]	✓	✗	Linear	✗	✓	selectively secure
[22]	✓	✓	none	✓	✗	fully secure
[31]	✓	✗	Linear	✗	✗	fully secure
[24]	✓	✗	none	✗	✗	fully secure
[25]	✓	✗	Linear	✗	✗	selectively secure
Ours	✓	✓	none	✓	✓	fully secure

Table 2. Storage and Transmission Overhead Comparison

Scheme	Key length	Ciphertext length
[36]	$(2+n)L_G$	$(2+3l+D)L_G+L_{G_T}$
[37]	$6L_G+(\lvert U\rvert+1)L_{Z_p}$	$(3+\lvert U\rvert)L_G+lL_{G_T}$
[22]	$(4+n)L_G+2L_{Z_p}$	$(4+2l)L_G+L_{G_T}$
[31]	$(3+n)L_G+2L_{Z_p}$	$(2+3l)L_G+L_{G_T}$
[24]	$(6+\lvert \mathrm{U}\rvert)L_G+L_{Z_p}$	$(3+2l)L_G+L_{G_T}$
[25]	$(3+2\lvert \mathrm{U}\rvert)L_G+L_{Z_p}$	$(1+4l)L_G+L_{G_T}$
Ours	$(6+n)L_G+2L_{Z_p}$	$(5+2l+D)L_G+L_{G_T}$

Table 3. Computational Complexity Comparison

Scheme	KeyGen.user	KeyGen.authority	Encryption	Decryption
[36]	✗	$(3+4\lvert \mathrm{S}\rvert)E_G$	$(2+5l+D)E_G+E_{G_T}$	$(2+3l)P+lE_{G_T}$
[37]	✗	$(4+\lvert \mathrm{S}\rvert)E_G$	$(2+2l+\lvert U\rvert)E_G+E_{G_T}$	$(3+2l)P+(2+l)E_G+lE_{G_T}$
[22]	E_G	$(6+\lvert \mathrm{S}\rvert)E_G$	$(4+3l)E_G+E_{G_T}$	$(3+2l)P+(3+l)E_G+lE_{G_T}$
[31]	✗	$(4+\lvert \mathrm{S}\rvert)E_G$	$(2+5l)E_G+E_{G_T}$	$(1+3l)P+(1+l)E_G+lE_{G_T}$
[24]	✗	$(11+\lvert \mathrm{S}\rvert)E_G$	$(3+2l)E_G+E_{G_T}$	$(2+2l)P+3E_G+(l+1)E_{G_T}$
[25]	✗	$(3+4\lvert \mathrm{S}\rvert)E_G$	$(1+4l)E_G+E_{G_T}$	$(1+4l)P+E_G$
Ours	E_G	$(7+\lvert \mathrm{S}\rvert)E_G$	$(5+3l+D)E_G+E_{G_T}$	$(5+2l)P+(3+l)E_G+lE_{G_T}$

6.3 Implementation

We provide the implementation of our scheme and other relevant schemes [22, 24, 25, 31, 36, 37]. Note that our scheme is built based on composite order pairings, but we can extend our scheme to prime order setting by using the techniques introduced in [16]. So we use PBC library to realize the prime order symmetrical bilinear pairing $e: G\times G\to G_T$ over the security level of 80 bits to implement the algorithm of the schemes in the **Table 3** in a simulation way. The hardware we used is R5-4800H with 8GB RAM. OS is windows 10 1909. As shown in **Fig. 6**, **Fig. 7** and **Fig. 8**, the evaluation includes KeyGen.authority complexity, decrypt complexity and encrypt complexity mainly.

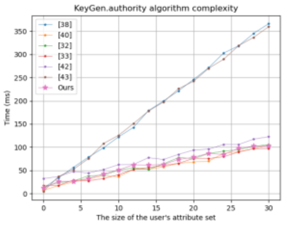


Fig. 6. KeyGen Authority Complexity

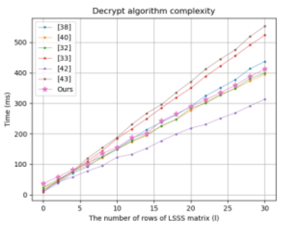


Fig. 7. Decrypt Complexity

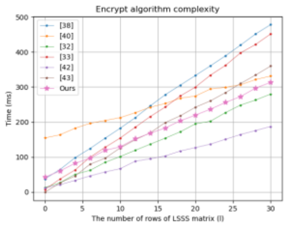


Fig. 8. Encrypt Complexity

As can be seen from **Fig. 6**, **Fig. 7** and **Fig. 8**, the computational complexity of the authority side in the KeyGen algorithm is much lower than that of schemes [36] and [25], and lower than [24], which is basically the same as [22, 31, 37]. The complexity of decryption is much lower than that of scheme [33, 43], and not much different from that of other schemes [22, 24, 36, 37]. The encryption complexity is much lower than that of scheme [31, 36, 37], and not much different from that of other schemes [22, 24, 25]. Therefore, in terms of efficiency, the proposed scheme has great advantages and competitiveness.

7. Conclusion

In this paper, we dispose the matters of key abuse and user revocation by introducing a CP-ABE scheme which is traceable user, accountable authority, and supports public auditing and user revocation. Furthermore, we demonstrate the scheme is full secure. Through theoretical analysis and implementation, we find our scheme only sacrifices a little time cost to achieve user revocation, which is a worthwhile compromise and has important significance in security and performance of this system.

Acknowledgement

This work was supported by the Natural Science Foundation of China under Grant 61520106007.

References

[1] H. Xiong, Y. Bao, X. Nie, and Y. I. Assor, “Server-aided attribute-based signature supporting expressive access structures for industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1013-1023, 2020. Article (CrossRef Link)

[2] H. Xiong, Q. Mei, and Y. Zhao, “Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIOT environments,” *IEEE Systems Journal*, vol. 14, no. 1, pp. 310-320, 2020. Article (CrossRef Link)

[3] T. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C. Chen, “An authenticated key exchange protocol for multi-server architecture in 5g networks,” *IEEE Access*, vol. 8, pp. 28096-28108, 2020. Article (CrossRef Link)

[4] T. Wu, C. Chen, K. Wang, C. Meng, and E. K. Wang, “A provably secure certificateless public key encryption with keyword search,” *Journal of The Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20-28, 2019. Article (CrossRef Link)

[5] H. Xiong, Y. Wu, C. Jin, and S. Kumari, “Efficient and Privacy-Preserving Authentication Protocol for Heterogeneous Systems in IIoT,” *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713-11724, Dec. 2020. Article (CrossRef Link)

[6] C. Chen, B. Xiang, Y. Liu, and K. Wang, “A secure authentication protocol for internet of vehicles,” *IEEE Access*, vol. 7, pp. 12047-12057, 2019. Article (CrossRef Link)

[7] H. Xiong, Y. Zhao, Y. Hou, X. Huang, C. Jin, L. Wang, and S. Kumari, “Heterogeneous Signcryption with Equality Test for IIoT environment,” *IEEE Internet of Things Journal*, p. 1, July 2020. Article (CrossRef Link)

[8] H. Hong, D. Chen, and Z. Sun, “A practical application of cp-abe for mobile phr system: a study on the user accountability,” *SpringerPlus*, vol. 1320, 2016. Article (CrossRef Link)

[9] H. H. Chung, P. S. Wang, T. Ho, H. Hsiao, and F. Lai, “A secure authorization system in phr based on cp-abe,” pp. 1-4, 2015. Article (CrossRef Link)

[10] F. Xhafa, J. Wang, X. Chen, J. K. Liu, J. Li, and P. Krause, “An efficient phr service system supporting fuzzy keyword search and fine-grained access control,” *Soft computing*, vol. 18, pp. 1795-1802, 2014. Article (CrossRef Link)

[11] F. Xhafa, J. Feng, Y. Zhang, X. Chen, and J. Li, “Privacy-aware attribute-based phr sharing with user accountability in cloud computing,” *The Journal of Supercomputing*, vol. 71, pp. 1607-1619, 2015. Article (CrossRef Link)

[12] L. Zhang, Q. Wu, Y. Mu, and J. Zhang, “Privacy-preserving and secure sharing of phr in the cloud,” *Journal of Medical Systems*, vol. 40, no. 267, 2016. Article (CrossRef Link)

[13] Z. Qin, Y. Wang, H. Cheng, Y. Zhou, Z. Sheng, and V. C. M. Leung, “Demographic information prediction: A portrait of smartphone application users,” *IEEE Transactions on Emerging Topics in Computing*, vol. 3494, pp. 432-444, 2018. Article (CrossRef Link)

[14] H. Xiong, J. Chen, Q. Mei, and Y. Zhao, “Conditional Privacy-Preserving Authentication Protocol with Dynamic Membership Updating for VANETs,” *IEEE Transactions on Dependable and Secure Computing*, p. 1, Dec. 2020. Article (CrossRef Link)

[15] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. pp. 457-473, 2005. Article (CrossRef Link)

[16] J. Li, K. Ren, and K. Kim, “A2be: Accountable attribute-based encryption for abuse free access control,” *IACR Cryptology ePrint Archive*, 2009. Article (CrossRef Link)

[17] 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. Article (CrossRef Link)

[18] 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,” in *Proc. of 2013 ACM SIGSAC Conference on Computer and Communications Security*, pp. 475-486, 2013. Article (CrossRef Link)

[19] J. Li, X. Chen, S. S. M. Chow, Q. Huang, D. S. Wong, and Z. Liu, “Multi-authority fine-grained access control with accountability and its application in cloud,” *Journal of Network and Computer Applications*, vol. 112, no. 15, pp. 89-96, 2018. Article (CrossRef Link)

[20] G. Yu, X. Ma, Z. Cao, W. Zhu, and J. Zeng, “Accountable multiauthority ciphertext-policy attribute-based encryption without key escrow and key abuse,” in *Proc. of International Symposium on Cyberspace Safety and Security*, pp. 337-351, 2017. Article (CrossRef Link)

[21] Z. Liu and D. S. Wong, “Practical attribute-based encryption: Traitor tracing, revocation and large universe,” *The Computer Journal*, vol. 59, no. 7, pp. 983-1004, 2016. Article (CrossRef Link)

[22] J. Ning, X. Dong, Z. Cao, and L. Wei, “Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud,” in *Proc. of European Symposium on Research in Computer Security*, pp. 270-289, 2015. Article (CrossRef Link)

[23] Z. Zhang, P. Zeng, B. Pan, and K. K. R. Choo, “Large-Universe Attribute-Based Encryption With Public Traceability for Cloud Storage,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10314-10323, Oct. 2020. Article (CrossRef Link)

[24] J. Ning, Z. Cao, X. Dong, and L. Wei, “White-Box Traceable CP-ABE for Cloud Storage Service: How to Catch People Leaking Their Access Credentials Effectively,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 883-897, 2016. Article (CrossRef Link)

[25] J. Zhao and P. Zeng, “Efficient, and Large Universe Ciphertext-Policy Attribute-Based Encryption with Black-Box Traceability for eHealth,” in *Proc. of the International Conference on Cyber Security Intelligence and Analytics*, vol. 1147, pp. 480-485, 2020. Article (CrossRef Link)

[26] M. Pirretti, P. Traynor, P. Mcdaniel, and B. Waters, “Secure attribute-based systems,” *Journal of Computer Security*, vol. 18, no. 5, pp. 799-837, 2010. Article (CrossRef Link)

[27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. of the 13th ACM Conference on Computer and Communications Security*, pp. 89-98, 2006. Article (CrossRef Link)

[28] X. Liang, R. Lu, X. Lin, and X. S. Shen, “Ciphertext policy attribute-based encryption with efficient revocation,” *TechnicalReport, University of Waterloo*, vol. 2, p. 8, 2010.

[29] Q. Liu, G. Wang, and J. Wu, “Time-based proxy re-encryption scheme for secure data sharing in a cloud environment,” *Information Sciences*, vol. 258, pp. 355-370, 2014. Article (CrossRef Link)

[30] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Proc. of Annual Cryptology Conference*, vol. 7417, pp. 199-217, 2012. Article (CrossRef Link)

[31] X. Yan, X. He, J. Yu, and Y. Tang, "White-box traceable ciphertext-policy attribute-based encryption in multi-domain environment," *IEEE Access*, vol. 7, pp. 128298-128312, 2019. Article (CrossRef Link)

[32] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. of Annual International Cryptology Conference*, vol. 2139, pp. 41-62, 2001. Article (CrossRef Link)

[33] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1442-1455, 2015. Article (CrossRef Link)

[34] Q. Mei, H. Xiong, J. Chen, M. Yang, S. Kumari, and M. K. Khan, "Efficient Certificateless Aggregate Signature With Conditional Privacy Preservation in IoV," *IEEE Systems Journal*, pp. 1-12, Feb. 2020. Article (CrossRef Link)

[35] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Proc. of Annual International Conference on the Therory and Applications of Cryptographic Techniques*, vol. 6110, pp. 62-91, 2010. Article (CrossRef Link)

[36] S. Xu, G. Yang, Y. Mu, and X. Liu, "A secure IOT cloud storage system with fine-grained access control and decryption key exposure resistance," *Future Generation Computer Systems*, vol. 97, pp. 284-294, 2019. Article (CrossRef Link)

[37] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Generation Computer Systems*, vol. 93, pp. 903-913, 2019. Article (CrossRef Link)

Wei Zhang received his master's degree in computer science from UESTC in 2014 and in 2016 continued to pursue doctor degree. His research interests include cryptography, blockchain and big data.

Yi Wu is currently a graduate student at the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC). Her research interests include cryptography and information security.

Hu Xiong received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009. He is currently a Full Professor with the School of Information and Software Engineering, UESTC. His research interests include public key cryptography and blockchain.

Zhiguang Qin is a national special grant winner, a member of the Computer Science and Technology Group of the sixth and Seventh discipline Review Group of the State Council, and a member of the national first-level discipline demonstration expert group on cyberspace security. His research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.