Blockchain-Assisted Proxy Re-Encryption Scheme With Revocation for Federal Diagnostics

Qingyang Zhang, Yu Wang, Jie Cui, *Senior Member, IEEE*, Bei Li, *Member, IEEE*, Jiaxin Li, and Hong Zhong

***Abstract*—Federated diagnosis, a concept emerging in Internet of Things (IoT)-based e-health systems, addresses the interconnectivity challenges of medical resources across different regions. Furthermore, flexible access control is required, which allows users to modify the access policy for encrypted data in the cloud without revealing sensitive information. Current solutions rely on third parties for policy modification, incur high computational overheads during user revocation, and expose user attributes to plaintext access policies. To address these issues, this study introduces a blockchain-assisted revocable federated diagnostic ciphertext policy attribute-based encryption (RFD-CPABE) scheme that enables policy conversion and revocation, which allows users to convert attribute-based encryption (ABE) ciphertext to inner product encryption (IPE) ciphertext swiftly and facilitates fast revocation using binary trees. This scheme enhances privacy protection by separating attribute names from values, thus concealing sensitive information within the access policies. Moreover, to reduce the computational burden on trusted institutions, the workload is decentralized utilizing a blockchain to minimize the pressure on the central servers. The formal security proof demonstrates the resilience of the scheme against selective chosen-plaintext attacks (CPA), and the experimental analysis confirms its superior efficiency compared to existing solutions.**

***Index Terms*—Access control, blockchain, federated diagnosis, Internet of Things (IoT), privacy-preserving, revocation.**

NOMENCLATURE

E_g	Exponentiation in G_1 and G_T.
P_g	Bilinear pairing computation.
M_g	Multiplication operation in G_1 and G_T.
l	Access policy size.
s	Initial attribute value count.
n	Number of attributes that match the policy.
r	Length of the $cover(R)$.
t	Maximum number of revoked users.
l'	Size of the revocation policy in [37].

Received 8 September 2025; revised 9 November 2025; accepted 6 December 2025. Date of publication 11 December 2025; date of current version 26 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62272002, Grant 62202005, Grant 62372002, and Grant 62572003; and in part by the Natural Science Foundation of Anhui Province, China, under Grant 2508085QF243. *(Corresponding author: Jie Cui.)*

Qingyang Zhang, Yu Wang, Jie Cui, Bei Li, and Hong Zhong are with the Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education and Anhui Engineering Laboratory of IoT Security Technologies, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn).

Jiaxin Li is with the Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei, Anhui 230039, China, and also with the Security Research Institute, New H3C Group, Hefei 230088, China (e-mail: li.jiaxin@h3c.com).

Digital Object Identifier 10.1109/JIOT.2025.3642872

I. INTRODUCTION

THE rapid development of the Internet of Things (IoT) has ushered in a new era of intelligent electronic medical care, which empowers efficient healthcare services by integrating medical devices and data [1], [2], [3]. E-health systems are superior to traditional paper-based methods. They enable the cloud-based storage and sharing of electronic medical records, thereby promoting the interconnectivity of medical resources across regions [4]. However, relying on third-party cloud providers raises concerns regarding data security and user privacy [5]. To address these concerns, attribute-based encryption (ABE) [6] has been proposed as a potential solution for protecting sensitive medical information. Although ABE provides effective access control mechanisms, its static nature and reliance on predefined access policies limit its applicability in scenarios that require dynamic collaboration with flexible policy modifications, such as federated diagnosis (FD).

We define FD as a case-specific, privacy-preserving, cross-institutional collaborative process in which multiple healthcare entities jointly produce a diagnosis or decision support for an individual patient without sharing raw data. In e-health systems, FD applies to scenarios such as remote second opinions and cross-hospital consultations for rare diseases at small institutions. As clinical contexts change, participating entities and their access to encrypted data may adjust dynamically, making access control inherently dynamic; therefore, systems must support fine-grained policy formulation, timely delegation, and efficient permission revocation.

The following scenario illustrates the limitations of traditional ABE schemes in supporting FD. As depicted in Fig. 1, Bob, a patient with lung disease, encrypts his medical examination results using an access policy P1: "Hefei," "Respiratory Medicine," and "Assistant Director Physician," thereby granting access only to the assistant director physician of the respiratory department in Hefei. The encrypted data and access policy are stored in the cloud. Alice, who meets access policy requirements, can access the information of Bob. However, when the condition of Bob worsened, he required medical resources in Beijing, to assist with his treatment. A

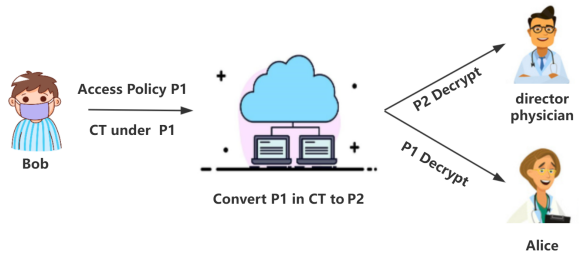


Fig. 1. Federal diagnostic scenario.

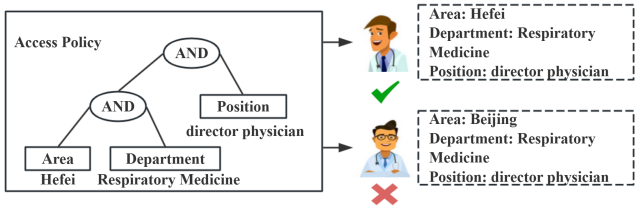


Fig. 2. Partial hiding policy example.

new access policy P2: "Beijing," "Respiratory Medicine," and "Director Physician" must be defined to enable secure access to the results of Bob by the Director Physician in Beijing for advanced treatment. If the condition of Bob improves subsequently, we must revoke the access rights of P2 to minimize treatment costs. The challenge is to efficiently modify access policies and revoke the access rights of users over time.

To address these challenges, existing methods utilize attribute-based proxy re-encryption (ABPRE). Existing ABPRE solutions typically rely on a trusted third party to generate re-encryption keys, thereby enabling the conversion of the ciphertext access policy from P1 to P2 in the cloud. This approach aims to reduce the computational burden on users and allows Bob to receive more advanced treatment. However, outsourcing computational tasks to a third party or relying on cloud services does not fundamentally address the issue of high computational overhead. Although the computational workload may shift because of this, it fails to reduce the overall resource consumption. Moreover, introducing a third-party entity to help with computation will incur additional operation and maintenance costs as well as communication overhead, which will significantly affect the practicality of the e-health system.

Furthermore, the existing ABPRE revocation schemes present significant challenges in terms of efficiency and privacy. When a user needs to be revoked, the re-encryption key must be regenerated, which places a substantial burden on a trusted third party. This entity is responsible for generating new re-encryption keys, identifying revoked users, and managing the entire key distribution process. Furthermore, the resulting overhead can hinder timely revocation and potentially affect system performance. Existing ABPRE schemes neglect the privacy risks associated with storing access policies in plaintext on the cloud. For instance, the inclusion of "Respiratory Medicine" in access structure P1 could compromise the medical information privacy of Bob by potentially revealing his lung disease.

A. Motivation

Although the existing ABPRE schemes provide data security and facilitate the sharing of encrypted data, they suffer from limitations in efficiency, revocation mechanisms, and user privacy protection. Therefore, this study introduces a more effective ABE scheme to support the sharing of encrypted data, protect user privacy, and provide direct revocation. This proposed ABE mechanism offers the following functionalities.

1) Enables efficient and lightweight access policy modification without third-party involvement. Users can modify access policies through simple calculations, thereby facilitating federal diagnosis.
2) No regeneration of re-encryption keys, thus enabling a prompt user revocation. This significantly reduces the workload of the trusted institutions and improves the revocation performance of the scheme.
3) Establishes a hidden access policy for sensitive privacy protection mechanisms during data interaction to ensure the confidentiality and integrity of the medical data.

B. Related Work

1) Attribute-Based Encryption: ABE [6] is generally divided into two types. Key-policy ABE (KP-ABE) [7] and ciphertext-policy ABE (CP-ABE) [8]. In CP-ABE, a user's key is associated with their attribute set, while the ciphertext contains the access structure. To enhance both the efficiency [9], [10], [11], [12] and security [13], [14] of ABE schemes, several enhancements have been proposed.

2) Policy-Hidden ABE: Although the performance and confidentiality of ABE have been significantly improved, the access structure still reveals user privacy information. To safeguard user privacy, several CP-ABE schemes incorporating access policy concealment have been introduced [15], [16], [17]. Nishide et al. [18] proposed partially hiding the strategy and allowing AND gates to appear in the access structure. Gu et al. [19] proposed an efficient ABE method for reliable policy updates under complete policy hiding. By dividing attributes into attribute names and values, Lai et al. [20] and Zhang et al. [21] implemented a form of partial policy concealment for access control policies. They only allow attribute names to appear in the access structure, as shown in Fig. 2 (similar methods can also be found in [22]). Inner product encryption (IPE) [23], [24], [23], an evolution of ABE, ingeniously achieves full access policy hiding by converting attributes into vectors. Although IPE is efficient, its expressiveness is limited to AND gates. Therefore, it is not as expressive as the linear secret sharing scheme (LSSS) [25].

3) ABE for Tracking and Revocation of Users: In addition to protecting user privacy, tracking and revoking users are equally important for a secure ABE system. Liu et al. [26] proposed a clear-box tracking scheme. Later, Ning et al. [27] extended the access structure in [26] by supporting the tracking of universe attribute sets. Ge et al. [28] implemented

revocation by modifying the access structure matrix in LSSS; however, this approach increased the decryption cost with each revocation operation. Wang et al. [29] used the user's personal information to build a binary tree and realize the revocation of user attributes. Li et al. [30] proposed an ABE scheme for revocable registration with user cancellation functionality.

4) Attribute-Based Proxy Re-Encryption: To facilitate data access for users with new access structures, proxy re-encryption [31], [32] was introduced. The concept of re-encryption was initially incorporated into attribute-based frameworks by Liang et al. [33], which facilitated a more adaptable representation of user identities. Following their contribution, a multitude of ABPRE schemes have emerged, with a primary focus on enhancing the expressiveness of access policies [34], [35], [36]. Ge et al. [37] addressed the user revocation problem in ABPRE; however, the computational overhead of their ABPRE is the same as the encryption and decryption overhead of ABE, which not only brought more computational overhead but also additional communication overhead.

5) Blockchain: Introduced by Nakamoto [38], blockchain's immutability and traceability properties offer significant advantages for simplifying complex computations in ABE schemes. Therefore, more and more ABE schemes have been combined with blockchain and widely used in medical systems [39]. Since hospitals do not trust each other, consensus nodes on the blockchain can be composed of various hospitals. These hospitals are subject to strict audits by the health department and use Byzantine fault-tolerant consensus to ensure data consistency even when a few nodes fail or act maliciously, thus ensuring the reliability of data management.

C. Our Contribution

To address the challenges of secure and efficient FD, we propose revocable federated diagnostic ciphertext policy ABE (RFD-CPABE), a novel blockchain-assisted ABE scheme. The main contributions of this study are as follows.

1) To address the high overhead of re-encryption key generation in ABPRE, we propose a novel access control scheme based on ABE and IPE. This scheme first encrypts the data using the highly expressive LSSS algorithm in ABE and then uses attribute vectors with high encryption efficiency in IPE to modify the access policy. This scheme guarantees consistent access semantics before and after data encryption, allowing data owners (DO) to quickly modify the access policy in the ciphertext stored in the cloud without interacting with an authority.
2) Considering that when revoking a user in ABPRE, the re-encryption key must be regenerated. A binary tree composed of users is used to complete the revocation, and the operation on the ciphertext is recorded in the blockchain to locate the user who must be revoked. Separating attribute names from values ensures that only the names are visible in the ABE access structure, while the values are securely embedded in the IPE vector, effectively hiding the policy of the user.

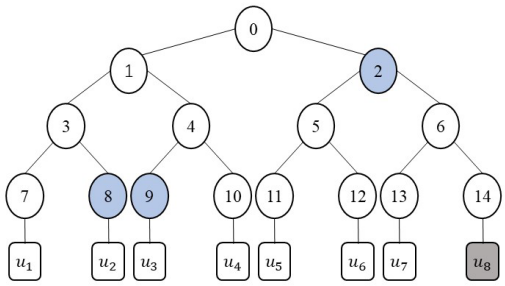


Fig. 3. Binary tree.

3) The formal security proof demonstrates the selective chosen-plaintext attack (CPA) security of the proposed scheme. Simulations and smart contract deployment in the blockchain demonstrate that the proposed scheme outperforms related approaches by reducing computational overhead. The feasibility of the practical application of the proposed scheme is also demonstrated.

II. Preliminaries

This section introduces the preparation work before this study.

A. Access Structure

Let $Y = \{Y_1, Y_2, \ldots, Y_n\}$ represent a set of participants (denoted as attributes in ABE). A monotone access structure $\mathbb{A}$ on Y is a nonempty subset of the power set 2^Y of Y, that is, $\mathbb{A} \subseteq 2^Y \setminus \{\emptyset\}$, and it satisfies the following monotonicity condition: for all B, C, if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. The sets in $\mathbb{A}$ are called the authorized set, while those not in $\mathbb{A}$ are considered unauthorized sets.

B. Binary Tree

Let UID represent the set of all users and Revoke represent the list of revoked users. Construct a binary tree Γ based on the user set.

1) Each leaf of the binary tree corresponds to a user uid in the system. Given that the total number of users is |UID|, the total number of nodes in the tree is $2|\text{UID}|-1$. Using a breadth-first search, all nodes are numbered starting from the root node, which is numbered 0, until the last leaf node is numbered $2|\text{UID}|-2$.
2) For any node i in Γ, Trail(i) represents the shortest path from the root node to node i.
3) Define DominatingSet(Revoke) as a minimum set of nodes that can cover all users not included in Revoke. For each unrevoked user uid, the intersection of its path Trail(uid) and DominatingSet(Revoke) must have one and only one node DistinctNode = DominatingSet(Revoke) $\cap$ Trail(uid).

As shown in Fig. 3, if the list Revoke is $\{u_1, u_4\}$, then the corresponding minimum cover set DominatingSet(Revoke) is $\{2, 8, 9\}$. In the tree, we know that the path Trail(u_8) of user u_8 is Trail(14) = $\{0, 2, 6, 14\}$. Therefore, the path intersection

of u_8 and DominatingSet(Revoke) contains only one node DominatingSet(Revoke) $\cap$ Trail(u_8) = {2}. The revocation is implemented utilizing the characteristics of the binary tree.

C. Inner Product Encryption (IPE)

Given a set of attribute values Attrv = $\{\text{attrv}_1, \text{attrv}_2, \ldots, \text{attrv}_n\}$ for the system.

1) For each access policy U composed of attribute subsets, it is transformed into an $n+1$-dimensional vector x, where $U \subseteq$ Attrv. The selection of components in the vector is as follows: for $1 \le i \le n$, $x_i \in Z_p^*$ is randomly chosen if $\text{attrv}_x \in U$ and $x_i = 0$ if $\text{attrv}_x \notin U$. Finally, $x_{n+1} = -\sum_{i=1}^{n} x_i (\text{mod } p)$.
2) For the attribute set SU submitted by the data user (DU), it is transformed into an $n+1$-dimensional vector y, where SU $\subseteq$ Attrv. The selection of components in the vector is as follows: for $1 \le i \le n$, $y_i = 1$ if $\text{attrv}_y \in$ SU, and $y_i = 0$ if $\text{attrv}_y \notin$ SU. Finally, $y_{n+i} = 1$.

When the attribute satisfies the access policy, $\langle x, y \rangle = 0$.

D. Complexity Assumptions

The hardness assumptions used to demonstrate the security of the proposed scheme are described.

Definition 1 (q-BDHE Assumption [22]): The description of the q-BDHE hardness assumption is as follows. Let G_1 be a cyclic group of prime order p under multiplication, where g is a generator of G_1. Define a bilinear map $e : G_1 \times G_1 \to G_T$. Let s and t be randomly selected elements from Z_p^*. Define a tuple R as $(g, g^s, g^t, g^{t^2}, \ldots, g^{t^q}, g^{t^{q+2}}, \ldots, g^{t^{2q}})$. There is no polynomial-time algorithm that can distinguish $e(g,g)^{t^{q+1}s}$ from a random element in G_T.

Definition 2 (DBDH Assumption): The description of the DBDH hardness assumption is as follows. Let G_1 be a cyclic group of prime order p under multiplication, where g is a generator of G_1. Define a bilinear map $e : G_1 \times G_1 \to G_T$. Let u, v, w, and z be randomly selected elements from Z_p^*. Define two tuples, R and R', as $(g^u, g^v, g^w, e(g,g)^{uvw})$ and $(g^u, g^v, g^w, e(g,g)^z)$. There is no polynomial-time algorithm that can distinguish between R and R'.

III. System and Security Models

A. System Architecture

Fig. 4 depicts the proposed system model architecture. The system consists of several interacting entities: DO, DA, DU, cloud server (CS), and the blockchain. Their tasks are introduced in detail below.

1) *DO:* Patients who share encrypted health information with hospitals for storage in the cloud. They can subsequently modify the access policy of the ciphertext as needed.
2) *DA:* The trusted third party is responsible for the management of doctor attributes and user groups and generates decryption keys for users. It can also locate users by accessing blockchain records.
3) *DU:* Data users are divided into two categories: doctors who meet the original access policy and doctors who meet the policy after it has been modified. Both types of doctors contribute to the federated diagnosis process.
4) *CS:* The patient's data is stored in the cloud, and the ciphertext can be modified according to the request of the DO. Using blockchain to record ciphertext operations.
5) *Blockchain:* This scheme adopts the alliance chain and selects large and reputable hospitals as consensus nodes. Smart contracts store the download, upload, and modification records of ciphertexts, thereby helping DA identify malicious users and helping CS locate ciphertexts for modification.

B. Formal Definition of the System

The RFD-CPABE scheme is formally constituted by the following seven algorithms, which are precisely defined as follows.

1) *Setup* $(\lambda, Attrv, \Gamma) \to PK, MSK$: The security parameter λ, the binary tree Γ of users, and a set of user attribute values Attrv are taken as input. This algorithm is run by DA, outputting the public key PK to be published to each entity and saving the system primary key MSK.
2) *Keygen (Attr, MSK)* $\to SK$: DA runs this algorithm. Input the attribute set Attr = (I_{Attrv}, Attrv) and the MSK. DA generates a decryption key SK related to the attribute set for the user.
3) *Encrypt (PK, MSP, Revoke,* $\mathbb{A}$*)* $\to CT$: This algorithm is run by DO and provides the information MSP to be encrypted, the PK, and the revocation list Recover. Define the access policy $\mathbb{A}$ according to the access requirements. Use this policy to encrypt the input message MSP and calculate the ciphertext CT. Send the generated CT containing the access policy $\bar{\mathbb{A}}$ of the deleted attribute value to the cloud.
4) *Decrypt (CT, SK)* $\to MSP$: DU provides its decryption key SK and CT containing the access policy of the hidden attribute value. Only DU whose attribute values and attribute values satisfy the access policy and are not listed in Revoke can decrypt MSP.
5) *Trace and Revoke (Rovke, CT)* $\to CT'$ *or* $\perp$: When a user needs to be revoked, DA uses the blockchain to help locate the revoked user and then checks whether the user exists in the user's binary tree. If it does not exist, the algorithm outputs $\perp$. If the user exists in Γ, Revoke is updated to NewRevoke and a new ciphertext CT$'$ is generated by the cloud.
6) *Share Encrypt (U)* $\to (CT^*)$: When a new doctor needs to access the data, the DO generates parameters TK based on the required access policy U and sends TK to the cloud. The CS will find the ciphertext to be modified based on the upload record of the ciphertext in the blockchain and modify it to CT*.
7) *Share Decrypt (CT***, SK)* $\to MSP$: The DU provides their SK and a ciphertext CT* which is stored in the cloud and uploaded by the attending physician. Only users whose attributes satisfy the access policy and are not listed in Revoke can decrypt MSP.

C. Security Model

In the proposed scheme, it is assumed that the CS is an honest but curious entity, meaning that the CS will follow the established protocol while attempting to collect all the information it can obtain. In our model, we assume that the DA and DO (patients) are fully trusted, meaning they will not collude with the CS. Furthermore, we assume that the blockchain will not access the ciphertext content or obtain the user's private information throughout the entire process. The proposed scheme employs two ciphertext formats: the initial ciphertext and a modified ciphertext resulting from access policy adjustments. According to the adversary's attack goal, we consider two selective CPA security games.

1) Game GAME-ABE: The selective CPA security model for the original ciphertext of this scheme is defined through a security game between a challenger C_{ABE} and an adversary $\mathrm{Adv}_{\mathrm{ABE}}$. The game is shown concretely as follows.

1) *Init:* Chooses an access structure $\mathbb{A}^* = (M^*, \pi^*, \Delta)$ and a revocation list Rover* as the challenge target of the $\mathrm{Adv}_{\mathrm{ABE}}$. In the access structure M^* is an $n^* \times m^*$ matrix, $\pi^*(i)$ maps the ith row in M^* to the attribute name. Moreover, $\Delta = \{\delta_{\pi^*(i)}\}_{i\in[1,n^*]}$ is the attribute value contained in $\pi^*(i)$ in the access structure.
2) *Setup:* C_{ABE} runs the Setup algorithm and sends the generated public key PK to the $\mathrm{Adv}_{\mathrm{ABE}}$.
3) *Phase 1:* $\mathrm{Adv}_{\mathrm{ABE}}$ uses a series of attribute sets $(\mathrm{uid}_1, \mathrm{Attr}_1), (\mathrm{uid}_2, \mathrm{Attr}_2), \ldots, (\mathrm{uid}_q, \mathrm{Attr}_q)$ to ask C_{ABE} for the corresponding decryption key.
   a) If Attr $\vDash \mathbb{A}^*$, it means that Attr satisfies the access structure $\mathbb{A}^*$. If uid $\notin$ Revoke*, then the algorithm terminates.
   b) If Attr $\nvDash \mathbb{A}^*$, it means that Attr not satisfies the access structure $\mathbb{A}^*$, or uid $\in$ Revoke*, C_{ABE} generates SK for $(\mathrm{uid}_i, \mathrm{Attr}_i)_{i\in[1,q]}$, and returns it to $\mathrm{Adv}_{\mathrm{ABE}}$.
4) *Challenge:* The $\mathrm{Adv}_{\mathrm{ABE}}$ submits two equal-length messages, M_0 and M_1, to the challenger C_{ABE}. Then, C_{ABE} randomly picks $M_\mu(\mu \in \{0,1\})$, then encrypts it using $\mathbb{A}^* = (M^*, \pi^*, \Delta)$, and Revoke* selected by $\mathrm{Adv}_{\mathrm{ABE}}$. Finally, the encrypted ciphertext is sent to $\mathrm{Adv}_{\mathrm{ABE}}$.
5) *Phase 2:* Phase 2 and Phase 1 are the same.
6) *Guess:* $\mathrm{Adv}_{\mathrm{ABE}}$ outputs a guess μ' for the value of μ. If the guess μ' is equal to the actual value μ (i.e., $\mu' = \mu$), then $\mathrm{Adv}_{\mathrm{ABE}}$ wins.

The advantage of $\mathrm{Adv}_{\mathrm{ABE}}$ winning the GAME-ABE is expressed as $\varepsilon = |\Pr[\mu = \mu'] - 1/2|$.

2) Game GAME-IPE: The selective CPA security model for the ciphertext after the access policy is modified can be described by a security game between a challenger C_{IPE} and an adversary $\mathrm{Adv}_{\mathrm{IPE}}$. This game is shown concretely as follows.

1) *Init:* $\mathrm{Adv}_{\mathrm{IPE}}$ chooses an access structure $\vec{a^*}$ to challenge.
2) *Setup:* C_{IPE} executes the Setup algorithm and sends the generated public key PK to the $\mathrm{Adv}_{\mathrm{IPE}}$.
3) *Phase 1:* $\mathrm{Adv}_{\mathrm{IPE}}$ uses a series of attribute sets $(\mathrm{uid}_1, \mathrm{Attr}_1), (\mathrm{uid}_2, \mathrm{Attr}_2), \ldots$ to ask C_{IPE} for the corresponding decryption key.
   a) If $\langle \vec{a^*}, \vec{b} \rangle = 0$, it means that Attr satisfies the access structure $\vec{a^*}$, the algorithm terminates.
   b) If $\langle \vec{a^*}, \vec{b} \rangle \neq 0$, it means that Attr not satisfies the access structure $\vec{a^*}$, C_{IPE} generates a decryption key of $(\mathrm{uid}_i, \mathrm{Attr}_i)$ and returns it to $\mathrm{Adv}_{\mathrm{IPE}}$.
4) *Challenge:* The $\mathrm{Adv}_{\mathrm{IPE}}$ submits two messages M_0 and M_1 of equal length to C_{IPE}. C_{IPE} randomly selects a message $M_\mu(\mu \in \{0,1\})$ and then calculate the new ciphertext using the access structure $\vec{a^*} = (a_1^*, a_2^*, \ldots, a_{n+1}^*)$ selected by $\mathrm{Adv}_{\mathrm{IPE}}$. Finally, the encrypted ciphertext is sent to $\mathrm{Adv}_{\mathrm{IPE}}$.
5) *Phase 2:* Phase 2 and Phase 1 are the same.
6) *Guess:* $\mathrm{Adv}_{\mathrm{IPE}}$ outputs a guess μ' for the value of μ. If the guess μ' is equal to the actual value μ (i.e., $\mu' = \mu$), then $\mathrm{Adv}_{\mathrm{IPE}}$ wins.

The advantage of $\mathrm{Adv}_{\mathrm{IPE}}$ winning the GAME-IPE is expressed as: $\varepsilon = |\Pr[\mu = \mu'] - 1/2|$.

Definition 3: Under a selective access policy, the RFD-CPABE scheme achieves against CPA, if GAME-ABE and GAME-IPE are secure.

IV. Proposed Scheme

This section introduces the RFD-CPABE scheme in detail. RFD-CPABE enables efficient modification of ciphertext access policies, supports rapid user revocation, and safeguards DO privacy with blockchain assistance.

A. Setup

Given the security parameter λ, the system's attribute value domain Attrv, and the binary trees Γ generated by all users, these are input to the DA. A prime order multiplicative cyclic group G_1 with generator g is selected, and a bilinear mapping $e : G_1 \times G_1 \to G_T$ is chosen. Set $\mathbb{G} = (G_1, G_T, p, g, e)$.

1) Randomly select $\gamma, \beta, \eta \in Z_p$ and $f_1, \ldots, f_{\mathrm{Attrv}} \in G_1$, then computer $e(g,g)^\gamma$ and g^β.
2) For the node in Γ, randomly choose $\{x_i\}_{i=0}^{2|\mathrm{UID}|-2} \in Z_p$, where |UID| indicates the total number of users. Then compute $\{y_i = g^{\eta x_i}\}_{i=0}^{2|\mathrm{UID}|-2}$.
3) Based on the defined quantity of attribute values |Attrv| within the system, DA randomly selects $\vec{c} = \{c_1, c_2, \ldots, c_{|\mathrm{Attrv}|+1}\} \in Z_p^*$ and computes $\{h_i = g^{c_i}\}_{i\in[1,|\mathrm{Attrv}|+1]}$.

The generated public key is $\mathrm{PK} = (\mathbb{G}, g^\beta, e(g,g)^\gamma, \{f_i\}_{i\in[\mathrm{Attrv}]}, \{y_i\}_{i=1}^{2|\mathrm{UID}|-2}, \{h_i\}_{i\in[1,|\mathrm{Attrv}|+1]})$ and save the primary key $\mathrm{MSK} = (\gamma, \beta, \eta, \{x_i\}_{i=0}^{2|U|-2}, \vec{c})$.

B. Keygen

The DU provides its attribute set $\mathrm{Attr} = (I_{\mathrm{Attrv}}, \mathrm{Attrv})$, where I_{Attrv} represents the attribute name in the access structure, $\mathrm{Attrv} = \{\mathrm{attrv}_i\}_{i\in I_{\mathrm{Attrv}}}$ and its unique uid to the DA. Assuming a $\mathrm{Tail}(i_0, \ldots, i_{\mathrm{uid}})$, where i_0 is the root and i_{uid} is a leaf node in Γ related to the DU. DA randomly selecting $r \in Z_p$ and then computed as follows:

$$\mathrm{SK}_1 = g^{\frac{r}{\eta x_{\mathrm{uid}}}}, \quad \left\{\mathrm{SK}_{2,i} = f_{\mathrm{attrv}_i}{}^r\right\}_{i\in I_{\mathrm{Attrv}}}$$
$$\mathrm{SK}_3 = g^\gamma g^{r\beta}, \quad \mathrm{SK}_4 = g^r.$$

DA computes the vector $\vec{b} = (b_1, b_2, \ldots, b_{|\text{Attrv}|+1})$ for each attribute value attrv_i belonging to the DU. For $1 \le i \le |\text{Attrv}|$, it computes

$$\begin{cases} b_i = 1, & \text{attrv}_i \in \text{DU} \\ b_i = 0, & \text{attrv}_i \notin \text{DU} \end{cases}$$

where $b_{|\text{Attrv}|+1} = 1$. Then, DA computes $\text{SK}_5 = g^{-r\langle \vec{c}, \vec{b} \rangle} g^{r\beta}$.

$\text{SK} = (\text{SK}_1, \{\text{SK}_{2,i}\}_{i \in I_{\text{Attrv}}}, \text{SK}_3, \text{SK}_4, \{x_i\}_{i \in \text{Tail}(i_{\text{uid}})}, \text{SK}_5)$ and $\{b_i\}_{i \in [1, |\text{Attrv}|+1]}$ are sent to the DU.

C. Encrypt

DO selects the private information MSP to be encrypted and encrypts it using the access structure $\mathbb{A} = (M, \pi, \Delta)$. Among them, M represents an $n \times m$ LSSS matrix, where n is the number of attribute names and π maps each row of M to an attribute name. $\Delta = \{\delta_{\pi(i)}\}_{i \in [1,n]}$ is the attribute value related with the access structure (M, π). The algorithm randomly selects a vector $\theta = (t, t_2, \ldots, t_n)^{\text{T}}$, where $t_2, \ldots, t_n \in Z_p$ and $t \in Z_p^*$. For each row $\{M_i\}_{i \in [1,n]}$ in M, compute $\lambda_i = M_i \cdot v$.

1) For $i \in [1, n]$, this algorithm randomly chooses $s_i \in Z_p$ and $a \in Z_p^*$. DO randomly selects $\vec{d} = \{d_1, d_2, \ldots, d_{|\text{Attrv}|+1}\} \in Z_p^*$. The following computations are performed:

$$C_0 = \text{MSP} \cdot e(g, g)^{\gamma t}, \quad C_1 = g^t$$
$$\left\{C_{i,1} = g^{\lambda_i \beta + a} f_{\delta_{\pi(i)}}{}^{-s_i}, C_{i,2} = g^{s_i}\right\}_{i \in [1,n]}$$
$$\left\{C_{i,3} = h_i^t g^{a+d_i}\right\}_{i \in [1, |\text{Attrv}|+1]}.$$

2) DS(Revoke) denotes the minimal cover set related to the revocation list Revoke. For each node $k \in \text{DS(Revoke)}$ in Revoke, the algorithm computes $\{T_k = y_k^a\}_{k \in \text{DS(Revoke)}}$ associated with the revocation list Revoke.
3) The incomplete access strategy of deleting attribute values is expressed as $\bar{\mathbb{A}} = (M, \pi)$. DO uses SHA_{256} to calculate the hash value of MSP for data identification, recorded as $\text{HD} = \text{SHA}_{256}(\text{MSP})$.

The DO stores $\vec{d}$ locally. Then, it uploads $\text{CT} = (C_0, C_1, \{C_{i,1}, C_{i,2}\}_{i \in [1,n]}, \{C_{i,3}\}_{i \in [1, |\text{Attrv}|+1]}, \{T_k\}_{k \in \text{DS(Revoke)}}, \text{Revoke}, \mathbb{A}^*, \text{HD})$ to the cloud. After receiving the ciphertext, the CS records (HD, DO_{id}, "upload") on the blockchain.

D. Decrypt

CS sends CT to the DU, while also uploading (HD, uid, "download") to the blockchain for recording. After receiving CT, the DU performs the following computations.

1) If the attribute set I_{Attrn} of the DU does not satisfy (M, π) or the user identity uid $\in$ Revoke, the algorithm is terminated.
2) If uid is not in Revoke and $I_{\text{Attrn}} \in (M, \pi)$.
   a) There exists a node $x_k \in \text{DS(Revoke)} \cap \text{Tail(uid)}$. The algorithm computes $\alpha = x_{\text{uid}}/x_k$ and then calculates $B = e(\text{SK}_1, T_k)^\alpha = e(g, g)^{ar}$.
   b) Assuming $V = \{i : \pi(i) \in \text{Attrn}\} \subseteq \{1, 2, \ldots, n\}$ that satisfies (M, π), there exist coefficients $\{w_i | i \in V\}$ such that $\sum_{i \in V} w_i M_i = (1, 0, \ldots, 0)$. Consequently, we can calculate $\sum_{i \in V} w_i \lambda_i = t$. The computation proceeds as follows:

$$\begin{aligned} D &= e(C_1, \text{SK}_3) = e(g, g)^{t\gamma} e(g, g)^{tr\beta} \\ E &= e\left(C_{i,1}, \text{SK}_4\right) e\left(\text{SK}_2, C_{i,2}\right) \\ &= e\left(g^{\lambda_i \beta + a} f_{\delta_{\pi(i)}}{}^{-s_i}, g^r\right) e\left(f_{\text{attrv}_i}{}^r, g^{s_i}\right) \\ &= e(g, g)^{ar} e(g, g)^{r\beta\lambda_i} \\ F &= \prod_{i \in I} \left(\frac{E}{B}\right)^{w_i} = \prod_{i \in I} \left(e(g, g)^{r\beta\lambda_i}\right)^{w_i} \\ &= e(g, g)^{r\beta t}. \end{aligned}$$

   c) Recover $\text{MSP} = C_0 F / D$.

E. Trace and Revoke

DA executes this algorithm, and DA can track the download records in the blockchain that need to be revoked. These records contain the unique uid of the user, then DA to check whether the uid exists in Γ. If the node corresponding to uid cannot be found in the leaf node of Γ, the algorithm terminates and outputs $\perp$.

If uid is found in the binary tree Γ, but does not exist in Revoke. Then, Revoke is updated to NewRevoke = Rrvoke $\cup$ {uid}. For k' in DS(NewRevoke), there are two cases.

1) If there is a $k \in \text{DS(Revoke)}$ such that $k = k'$, DA set $\xi_{k'} = 1$.
2) If there is a $k \in \text{DS(Revoke)}$ such that k is the ancestor of k', DA set $\xi_{k'} = x_{k'}/x_k$.

DA sends $\text{UpdateKey} = \{\xi_{k'}\}_{k' \in \text{DS(NewRevoke)}}$ to the cloud. To ensure the cloud can correctly update and revoke encrypted components related to users.

1) If there is $k \in \text{DS(Revoke)}$ such that $k = k'$, cloud calculate $T_{k'} = (T_k)^{\xi_{k'}}$.
2) If there exists $k \in \text{DS(Revoke)}$ such that k is the ancestor of k', cloud calculate $T_{k'} = (T_k)^{\xi_{k'}}$.

The CT be updated to $\text{CT}' = (C_0, C_1, \{C_{i,1}, C_{i,2}\}_{i \in [1,n]}, \{C_{i,3}\}_{i \in [1, |\text{Attrv}|+1]}, \{T_{k'}\}_{k' \in \text{DS(NewRevoke)}}, \text{NewRevoke}, \mathbb{A}^*, \text{HD})$.

F. Share Encrypt

1) *TransformKeygen:* When other doctors need to access patient data, the DO selects a new access structure $U \subseteq$ Attrv and establishes a vector $\vec{d} = (a_1, a_2, \ldots, a_{|\text{Attrv}|+1})$ of dimension $|\text{Attrv}| + 1$ based on the access structure. For $1 \le i \le |\text{Attrv}|$

$$\begin{cases} a_i \xleftarrow{R} Z_p^*, & \text{attrv}_i \in U \\ a_i = 0, & \text{attrv}_i \notin U \end{cases}$$

where $a_{|\text{Attrv}|+1} = -\sum_{i=1}^{|\text{Attrv}|} a_i (\text{mod } p)$. The DO calculates $\{C_i' = g^{a_i - d_i}\}_{i \in [1, |\text{Attrv}|+1]}$ as TK and sends it with DO_{id} to the cloud.
2) *UpdateCT:* The cloud finds the corresponding ciphertext through the DO_{id} and modifies the original ciphertext to $\text{CT}^* = (C_0' = C_0, C_1, \{C_{i,3} C_i'\}_{i \in [1, |\text{Attrv}|+1]}, \{T_k\}_{k \in \text{DS(Revoke)}}, \text{Revoke}, \text{HD})$ and modified the ciphertext in cloud, then use Algorithm 1 to record (HD, DO_{id}, "modify") on the blockchain.

Algorithm 1 RecordContract

```
RecordTable table = createRecordTable();
Entry entry = table.newEntry();
entry.set("Hash result of MSP", HD);
entry.set("The uid of the user", uid);
entry.set("Operation on the ciphertext", Operation);
entry.set("Timestamp", currentTimestamp());
entry.set("Previous Hash", getPreviousHash());
entry.set("Entry Hash", hashEntry(entry));
blockchain.insert(entry);
return entry;
```

G. Share Decrypt

CS sends CT^* to the DU, while uploading (HD, uid, "download") to the blockchain for recording. After receiving CT^*, the DU performs the following computations.

1) If the uid of DU in Revoke, the algorithm is terminated.
2) If uid $\notin$ Revoke.
   a) There exists a node $x_k \in \mathrm{DS(Revoke)} \cap \mathrm{Tail(uid)}$. This algorithm computes $\alpha = x_{\mathrm{uid}}/x_j$ and then calculates $B = e(\mathrm{SK}_1, T_k)^{\alpha} = e(g,g)^{ar}$.
   b) Then, DU performs the following computations:

$$D = e\,(C_1, \mathrm{SK}_3) = e(g,g)^{t\gamma} e(g,g)^{tr\beta}$$

$$L = e\left(\prod_{i=1}^{n+1} C_i' C_{i,3}{}^{b_i}, \mathrm{SK}_4\right) = e(g,g)^{rt\langle\vec{c},\vec{b}\rangle} e(g,g)^{r\langle\vec{a},\vec{b}\rangle} e(g,g)^{ra\sum_{i=1}^{|\mathrm{Attrv}|+1} b_i} = e(g,g)^{rt\langle\vec{c},\vec{b}\rangle} e(g,g)^{ra\sum_{i=1}^{|\mathrm{Attrv}|+1} b_i}$$

$$N = e\,(C_1, \mathrm{SK}_5) = e\left(g^t, g^{-r\langle\vec{c},\vec{b}\rangle} g^{r\beta}\right) = e(g,g)^{tr\beta} e(g,g)^{-tr\langle\vec{c},\vec{b}\rangle}.$$

   c) Recover $\mathrm{MSP} = C_0'\mathrm{NL/DB}^{\sum_{i=1}^{|\mathrm{Attrv}|+1} b_i}$.

V. Security Analysis

In this section, a formal security proof for the proposed scheme RFD-CPABE is provided.

Theorem 1: Aiming at the original ciphertext, the proposed scheme is selective CPA-secure if the q-BDHE hardness assumption holds, where $q > 2|\mathrm{UID}| - 1$ and $|\mathrm{UID}|$ is the number of users in the system.

Proof: The challenger C_{ABE} defines a bilinear map $e: G_1 \times G_1 \to G_T$, where G_1 is cyclic group of prime order p and g is the generator of group G_1. Then, C_{ABE} randomly selects a bit $\zeta \in \{0,1\}$. Give $R = (g, g^t, g^{\gamma}, g^{\gamma^2}, \ldots, g^{\gamma^q}, g^{\gamma^{q+2}}, \ldots, g^{\gamma^{2q}})$ and when $\zeta = 0$, then C_{ABE} computes $N = e(g,g)^{\gamma^{q+1}t}$; when $\zeta = 1$, C_{ABE} randomly selects $N \in G_T$.

Init: The $\mathrm{Adv}_{\mathrm{ABE}}$ selects an access structure $\mathbb{A}^* = (M^*, \pi^*, \Delta)$ to challenge, along with a revocation list Revoke^*. In this phase, M^* is a matrix of size $n^* \times m^*$, and π^* is a function that maps each row of M^* to a unique attribute name. The attribute values, denoted as $\Delta = \{\delta_{\pi^*(i)}\}_{i\in[1,n^*]}$ are associated with (M^*, π^*).

Setup: C_{ABE} runs the Setup algorithm and generated PK as follows.

1) Randomly choose $\gamma' \in Z_p$ and get $\gamma = \gamma' + d^{q+1}$ by setting $e(g,g)^{\gamma} = e(g,g)^{\gamma'} e(g^d, g^{d^q})$.
2) For Revoke^*, set $I_{\mathrm{Revoke}^*} = \{i \in \mathrm{Tail(uid)} | \mathrm{uid} \in \mathrm{Revoke}^*\}$ and randomly select $\{u_i \in Z_p\}_{\forall i=0,1,\ldots,2|\mathrm{UID}|-2}$. If $i \in I_{\mathrm{Revoke}^*}$, set $y_i = (g^{u_i} g^{d^i})^{\eta}$, then $x_i = u_i + d^i$; otherwise, set $y_i = (g^{u_i} g^{d^q})^{\eta}$, then $x_i = u_i + d^q$.
3) C_{ABE} simulation group elements $f_1, f_2, \ldots, f_{\mathrm{Attrv}}$. For each b in $1 \le b \le \mathrm{Attrv}$ begin by choosing a random value z_b, such that $\delta_{\pi^*(i)} = b$. f_b be set as $f_b = g^{z_b} \prod_{m=1}^{m^*} g^{d^m M_{k,n}^*}$, otherwise set $f_b = g^{z_b}$.

Send the follow public key PK to $\mathrm{Adv}_{\mathrm{ABE}}$:

$$\mathrm{PK} = \left(\mathbb{G}, e(g,g)^{\gamma}, g^{\beta}, \{f_i\}_{i\in[\mathrm{Attrv}]}\, \{y_i\}_{i=1}^{2|\mathrm{UID}|-2}, \{h_i\}_{i\in[1,|\mathrm{Attrv}|+1]}\right).$$

Phase 1: The $\mathrm{Adv}_{\mathrm{ABE}}$ requests a set of decryption keys for the corresponding attributes set $(\mathrm{uid}, \mathrm{Attr} = (I_{\mathrm{Attrv}}, \mathrm{Attrv}))$, where I_{Attrv} is the attribute name and $\mathrm{Attrv} = \mathrm{attrv}_{i\{i\in I_{\mathrm{Attrv}}\}}$ is the attribute value. There are four cases after submitting the attribute set.

Case 1: If $\mathrm{Attr} \vDash \mathbb{A}^*$, it means that Attr satisfies the access structure $\mathbb{A}^*$. If $\mathrm{uid} \notin \mathrm{Revoke}^*$, then the algorithm terminates.

Case 2: If $\mathrm{Attr} \vDash \mathbb{A}^*$ and $\mathrm{uid} \in \mathrm{Revoke}^*$, C_{ABE} performs the following computation.

1) Let $r = -d^q + d^{q-1} M_{i,1}^* / M_{i,2}^*$ and compute

$$\mathrm{SK}_{2,b} = (\mathrm{SK}_4)^{z_b} \left[\left(\prod_{k=1}^{m^*} g^{d^{q+k} M_{i,k}^*}\right)^{-1} \times \left(\prod_{k=1}^{m^*} g^{d^{q+k-1} M_{i,k}^*}\right)^{\frac{M_{i,1}^*}{M_{i,2}^*}}\right]$$

$$\mathrm{SK}_3 = g^{\gamma' + d^{q+1}} g^{\left(-d^q + d^{q-1}\frac{M_{i,1}^*}{M_{i,2}^*}\right)\beta}$$

$$\mathrm{SK}_4 = g^{\left(-d^q + d^{q-1}\frac{M_{i,1}^*}{M_{i,2}^*}\right)}.$$

2) Suppose that $\mathrm{Tail}(i_{\mathrm{uid}}) = \{i_0, \ldots, i_{\mathrm{uid}}\}$, where i_0 represents the root and i_{uid} corresponds to the leaf node value in the tree related to uid. Given that $\mathrm{uid} \in \mathrm{Revoke}^*$, then set $x_{i_{\mathrm{uid}}} = u_{i_{\mathrm{uid}}} + d^{i_{\mathrm{uid}}}$. C computes

$$\mathrm{SK}_1 = \left[g^{-d^q} g^{d^{q+1}\frac{M_{i,1}^*}{M_{i,2}^*}}\right]^{\frac{1}{(u_{\mathrm{uid}} + d^{\mathrm{uid}})\eta}}.$$

Case 3: If $\mathrm{Attr} \nvDash \mathbb{A}^*$ and $\mathrm{uid} \in \mathrm{Revoke}^*$, C_{ABE} performs the following computation.

1) A vector will be found that $\vec{w} = (w_1, w_2, \ldots, w_{n^*}) \in Z_p^{n^*}$ such that $w_1 = -1$ and we have $M_i^* \cdot \vec{w} = 0$ for all i where $\pi^*(i) \in I_{\mathrm{Attrv}}$.
2) Randomly select $s \in Z_p$ and let $r = s + w_1 d^q + w_2 d^{q-1} \cdot \cdots + w_{m^*} d^{q-n^*+1}$ and then compute

$$\mathrm{SK}_3 = g^{\gamma} g^{d^{q+1}} g^{\beta s} \prod_{j=1}^{m^*} g^{w_j d^{q-j+1}}$$

$$\mathrm{SK}_4 = g^{s} \prod_{j=1}^{m^*} g^{w_j d^{q-j+1}}.$$

If there exists b such that $\text{attrv}_i = \delta_{\pi^*(i)} = b$

$$\text{SK}_{2,b} = (\text{SK}_4)^{z_b} \cdot \prod_{j=1}^{m^*} \left(\left(g^{s \cdot d^j} \prod_{k=1}^{m^*} g^{w_k d^{q+1+j-k}} \right) \right)^{M^*_{i,j}}$$

otherwise $\text{SK}_{2,b} = (\text{SK}_4)^{z_b}$.

3) Suppose that $\text{Tail}(i_{\text{uid}}) = \{i_0, \ldots, i_{\text{uid}}\}$. Given that $\text{uid} \in \text{Revoke}^*$, then set $x_{i_{\text{uid}}} = u_{i_{\text{uid}}} + d^{i_{\text{uid}}}$. C_{ABE} computes

$$\text{SK}_1 = \left(g^s \prod_{j=1}^{m^*} g^{w_j d^{q+1-j}} \right)^{\frac{1}{(u_{\text{uid}} + d^{i_{\text{uid}}})\eta}}.$$

Case 4: If $\text{Attr} \nvDash \mathbb{A}^*$ and $\text{uid} \notin \text{Revoke}^*$, C_{ABE} performs the same computation as Case 3 and calculates SK_3, SK_4, and $\text{SK}_{2,b}$. Suppose that $\text{Tail}(i_{\text{uid}}) = \{i_0, \ldots, i_{\text{uid}}\}$. Given that $\text{uid} \notin \text{Revoke}^*$, then set $x_{i_{\text{uid}}} = u_{i_{\text{uid}}} + d^q$. C_{ABE} computes

$$\text{SK}_1 = \left(g^s \prod_{j=1}^{m^*} g^{w_j d^{q+1-j}} \right)^{\frac{1}{(u_{\text{uid}} + d^q)\eta}}.$$

Challenge: The Adv_{ABE} submits two messages of equal length, M_1 and M_2, to C_{ABE} and then calculates as follows.

1) C_{ABE} randomly picks up coin $\mu \in \{0, 1\}$ and then calculates $C_0 = M_\mu e(g, g)^{\gamma t}, C_1 = g^t, C_{i,2} = g^{-ad^i}$.
2) C_{ABE} randomly selects $r_2, \ldots, r_n \in Z_p^*$ and set $\theta = (t, td + r_2, td^2 + r_3, \ldots, td^{m^*-1} + r_{m^*})^{\text{T}}$ and compute $C_{i,1} = \prod_{j=2}^{m^*} (g^{\beta r_j})^{M^*_{i,j}} \prod_{j=1}^{m^*} (g^{\beta t d^{j-1}})^{M^*_{i,j}} (g^{z_{\delta_{\pi^*(i)}}})^{ad_i} g^a$.
3) For $\forall k \in \text{Revoke}(R^*)$, since $x_k = u_k + d^q$ and $y_k = g^{(u_k + d^q)\eta}$, then set $T_k = g^{(u_k + d^q)a\eta}$.

Phase 2: Phase 2 and Phase 1 are the same.

Guess: Adv_{ABE} provides a guess μ' for the value of μ. We consider the following two cases based on the relationship between μ and μ'.

1) If $\mu = \mu'$, C_{ABE} outputs a guessed $\xi = 0$ and $N = e(g, g)^{\alpha^{q+1} s}$. Assuming advantage of Adv_{ABE} is $\varepsilon = \Pr[\mu = \mu' | \xi = 0] - 1/2$, so the probability of C_{ABE} winning the game is $\Pr[\mu = \mu' | \xi = 0] = \varepsilon + 1/2$.
2) If $\mu \neq \mu'$, C_{ABE} outputs a guessed $\xi = 1$ and N is a random element in G_T. So, the advantage of Adv_{ABE} is $\varepsilon = \Pr[\mu \neq \mu' | \xi = 1] - 1/2 = 0$, the probability of C_{ABE} winning the game is $\Pr[\mu \neq \mu' | \xi = 1] = 1/2$.

Finally, the advantage of C_{ABE} in solving the q-BDHE assumption is as follows:

$$\text{Adv}_{C_{\text{ABE}}} = \left(\varepsilon + \frac{1}{2} \right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{2} \varepsilon.$$

Therefore, Theorem 1 is proved.

Theorem 2: Aiming at the ciphertext after modifying the access structure, the proposed scheme is selective CPA-secure if the DBDH hardness assumption holds.

Proof: The challenger C_{IPE} defines a bilinear map $e : G_1 \times G_1 \to G_T$, where G_1 is a cyclic groups of prime order p and g is the generator of group G_1. Then, C_{IPE} randomly selects a bit $\xi \in \{0, 1\}$. Give (U, V, W, Z) and when $\xi = 0$, then C_{IPE} computes $(U, V, W, Z) = (g^u, g^v, g^w, g^{uvw})$; when $\xi = 1$, $(U, V, W, Z) = (g^u, g^v, g^w, g^z)$, where $u, v, w \in Z_p^*$. Define system attribute value set Attrv, with the number of attribute values denoted as |Attrv| and the vector dimension is |Attrv|+1.

Init: The Adv_{IPE} chooses an access policy $\vec{a^*}$ to challenge.

Setup: C_{IPE} runs the Setup algorithm, computes $A = e(U, V) = e(g, g)^{uv}$, chooses random elements $c_1, c_2, \ldots, c_{|\text{Attr}+1|} \in Z_p^*$, and sets $h_i = g^{c_i}$. Finally, C_{IPE} sends PK to Adv_{IPE}

$$\text{PK} = \Big(\mathbb{G}, e(g, g)^\gamma, g^\beta, \{f_i\}_{i \in [\text{Attrv}]}, \{y_i\}_{i=1}^{2|\text{UID}|-2}, \{h_i\}_{i \in [1, |\text{Attrv}|+1]} \Big).$$

Phase 1: The Adv_{IPE} requests a set of decryption keys for the corresponding attributes set $(\text{uid}, \text{Attr} = (I_{\text{Attrv}}, \text{Attrv}))$, where I_{Attrv} is the attribute name and $\text{Attrv} = \text{attrv}_{i \{i \in I_{\text{Attrv}}\}}$ is the attribute value. C_{IPE} generates the corresponding private key SK

$$\text{SK} = \Big(\text{SK}_1, \{\text{SK}_{2,i}\}_{i \in I_{\text{Attrv}}}, \text{SK}_3, \text{SK}_4, \{x_i\}_{i \in \text{Tail}(i_{\text{uid}})}, \text{SK}_5, \{b_i\}_{i \in [1, |\text{Attrv}|+1]} \Big)$$

where $\langle \vec{a^*}, \vec{b} \rangle \neq 0$.

Challenge: The Adv_{IPE} submits two messages of equal length, M_1 and M_2, to C_{IPE}. C_{IPE} randomly picks up coin $\mu \in \{0, 1\}$ and then calculates $\text{CT}^* = (C'_0 = M_\mu Z, C_1, \{C_{i,3} C'_i\}, \{T_k\}_{k \in \text{DS}(\text{Revoke})}, \text{Revoke}, \text{HD})$.

1) If $\xi = 0$, $Z = e(g, g)^{uvw}$. Assume $w = t$, the ciphertext is as follows:

$$\text{CT}^* = \left(C'_0 = M_\mu e(g, g)^{uvw}, C_1, \{C_{i,3} C'_i\}, \{T_k\}_{k \in \text{DS}(\text{Revoke})}, \text{Revoke}, \text{HD} \right).$$

2) If $\xi = 1$, $Z = e(g, g)^z$. Then, the ciphertext is as follows:

$$\text{CT}^* = \left(C'_0 = M_\mu e(g, g)^z, C_1, \{C_{i,3} C'_i\}, \{T_k\}_{k \in \text{DS}(\text{Revoke})}, \text{Revoke}, \text{HD} \right).$$

Here, z is selected from Z_p^* randomly.

Phase 2: Phase 2 and Phase 1 are the same.

Guess: Adv_{IPE} provides a guess μ' for the value of μ. We consider the following two cases based on the relationship between μ and μ'.

1) If $\mu = \mu'$, C_{IPE} outputs a guessed $\xi = 0$ and $Z = e(g, g)^{uvw}$. Assuming advantage of Adv_{IPE} is $\varepsilon = \Pr[\mu = \mu' | \xi = 0] - 1/2$, so the probability of C winning the game is $\Pr[\mu = \mu' | \xi = 0] = \varepsilon + 1/2$.
2) If $\mu \neq \mu'$, C_{IPE} outputs a guessed $\xi = 1$ and Z is a random element in G_T. The advantage of Adv_{IPE} is $\varepsilon = \Pr[\mu \neq \mu' | \xi = 1] - 1/2 = 0$, and the probability of C_{IPE} winning the game is $\Pr[\mu \neq \mu' | \xi = 1] = 1/2$.

Finally, the advantage of C_{IPE} in solving the DBDH assumption is as follows:

$$\text{Adv}_{C_{\text{IPE}}} = \left(\varepsilon + \frac{1}{2} \right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{2} \varepsilon.$$

Therefore, Theorem 2 is proved.

VI. Efficiency Analysis

In this section, we evaluate and compare the functionality and efficiency of the proposed scheme with other schemes from both theoretical and experimental analysis perspectives.

TABLE I
EFFICIENCY COMPARISON

Scheme	Keygen	Encrypt	DecryptCT	Revoke
TR-AP-CPABE [22]	$(6+s)E_g+(1+s)M_g$	$(3+4l+r)E_g+(1+l)M_g$	$(2+3n)P_g+(3+n)E_g+(5+2n)M_g$	tE_g
CP-ABPRE-DR [38]	$(3+s)E_g+M_g$	$(3+3l)E_g+(1+l)M_g$	$(1+2n)P_g+nE_g+(2+n)M_g$	$(2+3(l+\bar{l}))E_g+(2+3(l+\bar{l}))M_g$
Ours	$(6+s)E_g+2M_g$	$(4+s+3l+r)E_g+(2+l+s)M_g$	$(2+2n)P_g+(1+n)E_g+(2n+2)M_g$	tE_g

TABLE II
EFFICIENCY COMPARISON IN MODIFYING ACCESS POLICIES

Scheme	TransformKeygen	UpdateCT	DecryptCT*
CP-ABPRE-DR [38]	$(6+s+3l^{'})E_g+(2+l^{'})M_g$	$(2+2l)P_g+lE+(2+l)M_g$	$(1+2n)P_g+(n+1)E_g+(3+n)M_g$
Ours	$(s+1)E_g$	$(s+1)M_g$	$4P_g+(s+2)E_g+(s+5)M_g$

A. Theoretical Complexity

In our theoretical analysis, the Nomenclature lists the symbols and their interpretations.

Table I compares the costs of Keygen, Encrypt, Decrypt, and Revoke between our scheme, [22], and [37]. In Keygen, our scheme requires fewer multiplications than [22] but slightly more than [37]. In Encrypt, efficiency is slightly lower than [22] and [37] due to embedding the IPE parameter in the ABE ciphertext. In Decrypt, fewer pairing operations than [22] yield efficiency comparable to [37] and better than [22]. For user tracking, blockchain records uid, removing the need for key integrity checks as in [22]. Revocation in [37] requires modifying re-encryption keys and updating ciphertexts, incurring high cost. Compared to the binary tree revocation algorithm in [22], we have improved the algorithm's correctness by ensuring that only users whose data has not been revoked can calculate $e(g,g)^{ar}$ using the binary tree. Our scheme modifies the ciphertext update method proposed by Han et al. [22] for better security while maintaining similar efficiency.

Our scheme and [37] both implement the function of modifying access policies. Table II shows the efficiency comparison between our solution and [37] in modifying access policies. The TransformKeygen algorithm in [37] has the same overhead as its encryption. In contrast, our scheme uses IPE encryption, resulting in lower overhead compared to ABE-based approaches. In our scheme, since there is no third party to assist in the computation, after DO generates the transform key, the cloud can find the corresponding ciphertext through the blockchain and modify it, so the ciphertext can be updated faster. The decryption of the updated ciphertext is more efficient than the ABE decryption in [37] since the use of IPE decryption.

B. Experimental Performance

1) To demonstrate the effectiveness of the proposed scheme, a series of experimental analyses is conducted. The experiments are performed on Ubuntu 22.04.4 64-bit, with 4 GB RAM, using Python 3.10 with the SS512 elliptic curve from the Charm 0.50 framework.

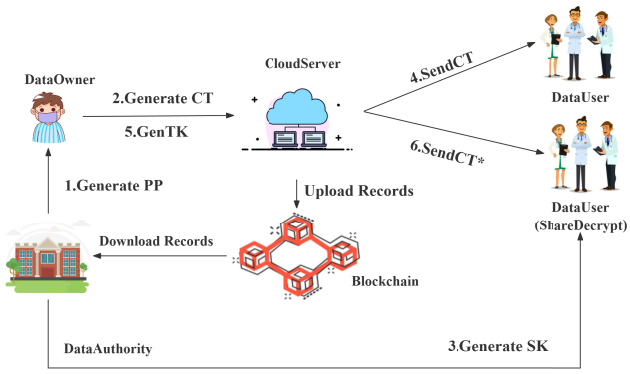


Fig. 4. System model.

The virtual machine is configured on an Intel Core i7-10700K CPU @2.9 GHz. It is equipped with 32 GB of RAM. We set the number of attribute values $|S| = 1000$ and all access strategies to full AND gates. Access policy size and re-encrypted access policy size ranged from 10 to 50 in steps of 10. The user attributes and re-encrypted user attributes are also set to range from 10 to 50 with a step size of 10. The user set was UID = $\{\text{uid}_1, \text{uid}_2, \text{uid}_3, \text{uid}_4, \text{uid}_5, \text{uid}_7, \text{uid}_8\}$, and the revocation list was $R = \{\text{uid}_6\}$. Each experiment is repeated 50 times to obtain the average time. Fig. 5 compares the execution times of our scheme with [22] and [37]. In Fig. 5(a), Keygen is slightly less efficient than [37] (no policy hiding) but significantly faster than [22] with the same functionality. Fig. 5(b) shows encryption cost grows linearly with attributes; ours is slightly higher than [22] and [37]. In Fig. 5(c), our scheme reduces about one-third of pairings versus [22], achieving decryption speed close to [37] and better than [22]. For revocation, ciphertext update time for uid_8 matches [22] at 1.6 ms. Fig. 6 demonstrates the comparison between our scheme and [37] in the process of modifying access policies. In Fig. 6(a), we analyze the computational cost of TransformKeygen, where our scheme outperforms [37]. For 50 attributes, our computation time is

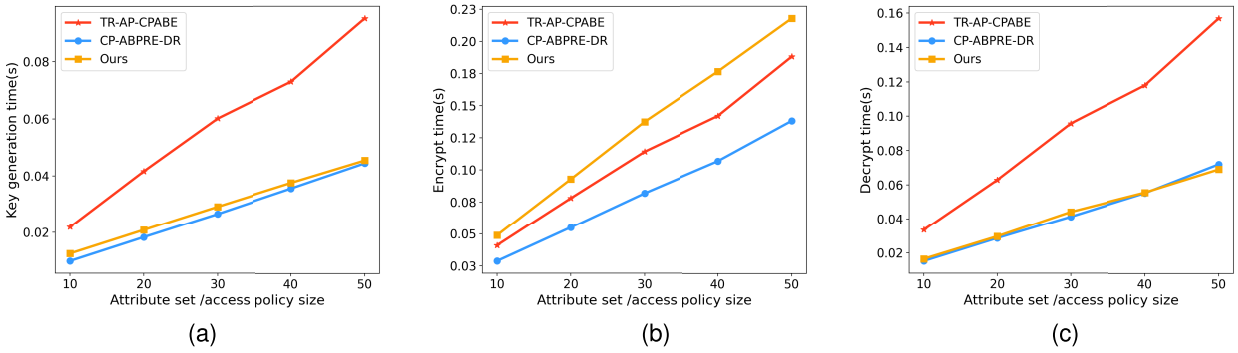


Fig. 5. Computation time of our proposed RFD-CPABE scheme. (a) Keygen time analysis. (b) Encrypt time analysis. (c) DecryptCT time analysis.

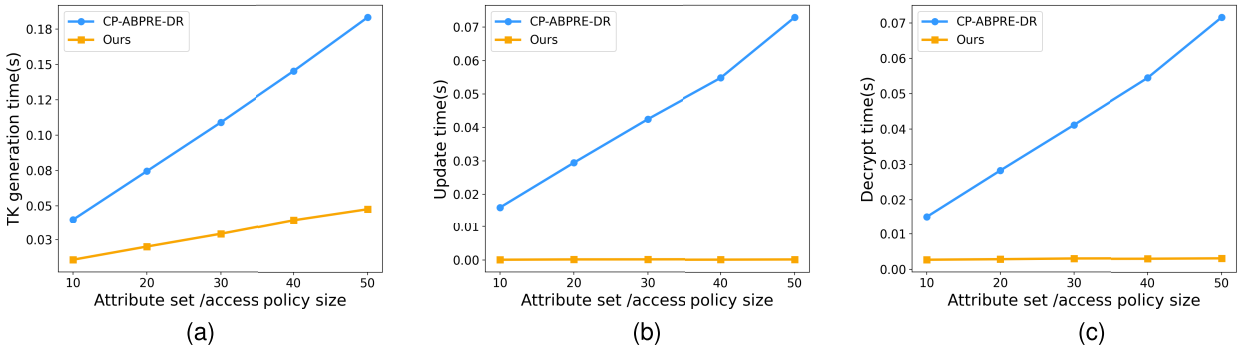


Fig. 6. Computation time of our proposed RFD-CPABE scheme in modifying access policies. (a) TransformKeygen time analysis. (b) UpdateCT time analysis. (c) DecryptCT* time analysis.

TABLE III
SMART CONTRACT SETTINGS AND PARAMETERS

Name	Value
Environment	Remix VM(Cancun)
Gas Limit	3000000
Account	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Smart Contract Address	0xd8b934580fcE35a11B58C6D73aDeE468a2833fa8

0.049 s, while [37] takes 0.18 s. Moreover, our scheme allows DO to modify the access policy, unlike [37], which allows DU to modify the access policy. Fig. 6(b) shows the comparison of the cost of updating ciphertext in the cloud between our scheme and [37]. With 50 attributes, our time is 2.7 ms, while [37] takes 72 ms. The comparison of decryption efficiency between our scheme and [37] after modifying the ciphertext is shown in Fig. 6(c). As the number of attributes increases, our decryption efficiency significantly improves compared to [37]. When there are 50 attributes, our decryption time is only 3 ms, while [37] takes 71 ms.

2) To evaluate our functionality for user tracking, we use Geth 1.10.25 to build a blockchain to test the gas cost of user upload records. Smart contracts are written in Solidity 0.8.7 using Remix IDE, and their design details are shown in Table III. In the experiment, uploading user operation records requires 405 443 gas, and the total transaction cost for the smart contract is 489 745 gas.

The experimental results demonstrate the efficacy and practicality of our proposed scheme. Based on IPE and ABE, our scheme achieves efficient ciphertext access policy modification, user revocation, and policy hiding. The blockchain is introduced to supervise user records and help the trusted institution locate users. This approach not only reduces computational costs but also protects user privacy, thereby realizing the objectives of secure federated diagnosis.

VII. CONCLUSION

In this study, we designed a blockchain-assisted access control scheme to realize federated diagnostic functions in the cloud, which realizes flexible data sharing and fast user revocation, significantly reduces the computing cost of patients and doctors, and hides the access policy during the interaction process. Patients can convert an ABE ciphertext in the cloud into an IPE ciphertext through simple computations, which reduces the computing cost of all entities while ensuring security. User revocation was streamlined using a binary tree to eliminate the need for complex calculations. Furthermore, the workload of trusted institutions can be reduced by leveraging smart contracts to record data operations. Formal security proofs and theoretical experimental analyses demonstrate that our scheme is efficient while ensuring security, which also demonstrates its applicability in medical systems.

ACKNOWLEDGMENT

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this article.

REFERENCES

[1] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[2] Q. Zhang, X. Zhou, H. Zhong, J. Cui, J. Li, and D. He, "Device-side lightweight mutual authentication and key agreement scheme based on chameleon hashing for industrial Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 7895–7907, 2024.

[3] S. Li, L. D. Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jan. 2018.

[4] X. Zhou et al., "Hierarchical federated learning with social context clustering-based participant selection for Internet of Medical Things applications," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 4, pp. 1742–1751, Aug. 2023.

[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan. 2012.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Adv. Cryptol. EUROCRYPT 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, May 2005, pp. 457–473.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct. 2006, pp. 89–98.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[9] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Inf. Sci.*, vol. 470, pp. 175–188, Jan. 2019.

[10] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proc. Int. Workshop Public Key Cryptogr.*, Feb. 2013, pp. 162–179.

[11] J. Cui, B. Li, H. Zhong, Y. Xu, and L. Liu, "Achieving revocable attribute group-based encryption for mobile cloud data: A multi-proxy assisted approach," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 2988–3001, Jul. 2023.

[12] W. Weng, J. Li, Y. Zhang, Y. Lu, J. Shen, and J. Han, "Efficient registered attribute based access control with same sub-policies in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 24, no. 9, pp. 8441–8453, Sep. 2025.

[13] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Proc. Annu. Cryptol. Conf.*, Aug. 2012, pp. 180–198.

[14] Q. Zhang, C. Xu, H. Zhong, C. Gu, and J. Cui, "Revocable and efficient blockchain-based fine-grained access control against EDoS attacks in cloud storage," *IEEE Trans. Comput.*, vol. 73, no. 8, pp. 2012–2024, Aug. 2024.

[15] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Proc. Int. Conf. Provable Secur.*, 2016, pp. 19–38.

[16] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 35–45, Jan. 2016.

[17] J. Li, E. Zhang, J. Han, Y. Zhang, and J. Shen, "PH-MG-ABE: A flexible policy-hidden multigroup attribute-based encryption scheme for secure cloud storage," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 2146–2157, Jan. 2025.

[18] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*. Berlin, Germany: Springer, 2008, pp. 111–129.

[19] C. Gu, J. Li, Y. Zhang, Y. Lu, and J. Shen, "EABE-PUFPH: Efficient attribute-based encryption with reliable policy updating under full policy hiding," *IEEE Trans. Comput.*, vol. 74, no. 11, pp. 3750–3762, Nov. 2025.

[20] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur.*, 2012, pp. 18–19.

[21] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[22] D. Han, N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 316–327, Jan. 2022.

[23] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2008, pp. 146–162.

[24] H. Wee, "Attribute-hiding predicate encryption in bilinear groups, revisited," in *Proc. Theory Cryptogr. Int. Conf.*, Baltimore, MD, USA, 2017, pp. 206–233.

[25] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 53–70.

[26] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 76–88, Jan. 2013.

[27] 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 Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 883–897, Sep. 2018.

[28] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2864–2872, Sep. 2022.

[29] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203225.

[30] J. Li, S. Chen, Y. Lu, J. Ning, J. Shen, and Y. Zhang, "Revocable registered attribute-based encryption with user deregistration," *IEEE Internet Things J.*, vol. 12, no. 15, pp. 31526–31535, Aug. 2025.

[31] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1998, pp. 127–144.

[32] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006.

[33] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur.*, Mar. 2009, pp. 276–286.

[34] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.

[35] K. Liang et al., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, Nov. 2015.

[36] Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in IIoT," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 6, pp. 5797–5809, Nov. 2024.

[37] C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo, and L. Fang, "Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 2, pp. 949–960, Mar. 2024.

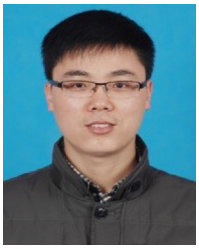
[38] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008, Art. no. 21260. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[39] F. Li, K. Liu, L. Zhang, S. Huang, and Q. Wu, "EHRChain: A blockchain-based EHR system using attribute-based and homomorphic cryptosystem," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2755–2765, Sep. 2022.

Qingyang Zhang was born in Anhui, China, in 1992. He received the B.Eng. and Ph.D. degrees in computer science from Anhui University, Hefei, China, in 2014 and 2021, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. He has more than 30 scientific publications in reputable journals (e.g., PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and IEEE TRANSACTIONS ON COMPUTERS) and international conferences. His research interest includes edge computing, computer systems, and security.

Yu Wang is currently a Research Student with the School of Computer Science and Technology, Anhui University, Hefei, China. His research focuses on the security of the Industrial Internet of Things.



Jie Cui (Senior Member, IEEE) was born in Henan, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2012.

He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. He has more than 150 scientific publications in reputable journals (e.g., IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON MULTIMEDIA), academic books, and international conferences. His current research interests include applied cryptography, IoT security, vehicular ad hoc networks, cloud computing security, and software-defined networking (SDN).

Bei Li (Member, IEEE) is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His research interests include IoT security, blockchain, and applied cryptography.

Jiaxin Li received the master's degree from Anhui University, Hefei, China, in 2020, where he is currently pursuing the Ph.D. degree.

He is currently holds the position of Director of Government Affairs with H3C Information Security Technology Company Ltd., Hefei. His research focuses on the security of vehicular ad hoc network, data security, and the security of Industrial Internet of Things.

Hong Zhong was born in Anhui, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005.

She is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. She has more than 200 scientific publications in reputable journals (e.g., IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON BIG DATA), academic books, and international conferences. Her research interests include applied cryptography, IoT security, vehicular ad hoc networks, cloud computing security, and software-defined networking (SDN).