Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.046106

ARTICLE

A Blockchain and CP-ABE Based Access Control Scheme with Fine-Grained Revocation of Attributes in Cloud Health

Ye Lu[1,*], Tao Feng[1], Chunyan Liu[2] and Wenbo Zhang[3]

[1]School of Computer & Communication, Lanzhou University of Technology, Lanzhou, 730000, China

[2]School of Economics & Management, Lanzhou University of Technology, Lanzhou, 730000, China

[3]School of Computer Science, Baoji University of Arts and Science, Baoji, 721000, China

*Corresponding Author: Ye Lu. Email: luye@lut.edu.cn

Received: 19 September 2023 Accepted: 16 November 2023 Published: 27 February 2024

ABSTRACT

The Access control scheme is an effective method to protect user data privacy. The access control scheme based on blockchain and ciphertext policy attribute encryption (CP–ABE) can solve the problems of single—point of failure and lack of trust in the centralized system. However, it also brings new problems to the health information in the cloud storage environment, such as attribute leakage, low consensus efficiency, complex permission updates, and so on. This paper proposes an access control scheme with fine-grained attribute revocation, keyword search, and traceability of the attribute private key distribution process. Blockchain technology tracks the authorization of attribute private keys. The credit scoring method improves the Raft protocol in consensus efficiency. Besides, the interplanetary file system (IPFS) addresses the capacity deficit of blockchain. Under the premise of hiding policy, the research proposes a fine-grained access control method based on users, user attributes, and file structure. It optimizes the data-sharing mode. At the same time, Proxy Re-Encryption (PRE) technology is used to update the access rights. The proposed scheme proved to be secure. Comparative analysis and experimental results show that the proposed scheme has higher efficiency and more functions. It can meet the needs of medical institutions.

KEYWORDS

Blockchain; access-control; CP-ABE; cloud health

1 Introduction

The rapid development of the Internet of Things and cloud computing enables wearable devices to collect and analyze human health data in real-time, including body temperature, blood oxygen, pulse, etc., by sharing the above data, combined with hospital examination records, it can provide a more accurate basis for doctors' diagnosis, which is a typical application scenario of cloud health.

CP-ABE [1] is a one-to-many access control policy widely adopted by cloud computing. Although many researchers have made improvements, there are still some problems: a single point of failure, such as property server downtime or attribute agencies maliciously divulging users' private keys;

Information leaks, such as the access policy: "(Doctor-ID: 1234) AND (Department: Infectious diseases) AND (All-data)", a malicious user can easily speculate that the patient has an infectious disease, obviously revealing the patient's privacy; The update of the access policy is complex or even missing, for example, when a medical research institution wants to obtain the patient's heart rate data and the blood oxygen data collected by the bracelet, the patient may not be able to change the policy to "(Disease research institutes) AND (Blood oxygen) AND (Heart rate)"; In addition, when the encrypted file corresponding to the access policy has a hierarchical relationship, for example, the medical data file contains the patient's name, as well as Blood oxygen data and Heart rate data. Patients want their attending doctors to access all the data, while research institutions can only access the examination records. The existing CP-ABE needs to encrypt each file separately, which will result in large computing and storage overhead.

The proposal of Hidden Policy CP-ABE (HP-CP-ABE) has improved the problem of user privacy disclosure. Literature [2] proposed a method to resist attribute value guessing attack (AVGA) and supports an outsourced decryption test algorithm to assist users in decrypting the ciphertext. However, this scheme does not support attribute revocation and ciphertext search.

The core advantage of blockchain technology is publicly verifiable, which can ensure the compliant use of medical data using digital signature, timestamp, distributed consensus, etc., thus providing support for solving the problems of single point failure, private key disclosure, inefficient and insecure data storage, etc. Although these characteristics of the blockchain are suitable to solve the inherent problems of the traditional CP-ABE single-attribute authority, the blockchain has the problems of a large amount of consensus calculation and high delay, such as proof-of-work (POW), Proof of Stake (POS), and so on. Although the efficiency of the Raft consensus mechanism is greatly improved, when there are many electors, the main algorithm cannot reach an agreement quickly, nor is it suitable for multi-node consensus. In addition, there is a problem of capacity expansion in the blockchain, the IPFS can reduce the storage load of the blockchain nodes and facilitate the retrieval of information.

To solve the above problems, this paper proposes a searchable fine-grained ciphertext access control policy based on blockchain and CPABE. The main contributions of this paper are as follows and the main functional differences from other access control schemes are shown in Table 1. Our scheme implements finer-grained access control and permission updates as well as attribute key traceability. Note that "✓" means this function is supported, while "×" means it is not supported.

Table 1: Main functional differences

Solutions		Ours	[2]	[3]	[4]	[5]	[6]	[7]
Fine grain size	User	✓	×	✓	✓	✓	✓	✓
	File	✓	×	×	×	×	×	×
	Attribute	✓	✓	×	✓	✓	✓	✓
Permission update	Attribute private key update	✓	×	×	✓	×	×	×
	Symmetric key update	✓	×	×	×	×	×	
	Key ciphertext update	✓	×	×	×	✓	×	✓
Traceability	Attribute private key distribution	✓	×	×	×	×	✓	×

1) Under the premise of policy hiding, structured data & user attribute permission tree is the first designed to support fine-grained access control in three dimensions: user, user attribute, and

file structure, to enhance privacy. The calculation complexity of the corresponding n files is O (n) ∗ symmetric encryption cost +O(1) ∗ asymmetric encryption cost, while the encryption cost of the non-file tree structure is o (n) ∗ asymmetric. Our solution significantly reduces computational complexity.

2) To update the attributes of the user, we have for the first time realized the symmetric key (used to encrypt a single file) update and the attribute private key (used to encrypt symmetric keys) update, as well as the attribute ciphertext synchronous update by PRE (see Func8–11). PRE is used to update the ciphertext of files on IPFS. As far as we know, none of the other schemes can achieve key updates.

3) The scheme uses blockchain to record the distribution of attribute keys and the storage address of patient information in IPFS. The traceability of key distribution is realized while solving the problem of insufficient blockchain capacity, and the improved Raft algorithm also speeds up the master selection rate and further improves the consensus rate.

The rest of this paper is organized as follows: Section 2 introduces relevant research progress. Section 3 gives background knowledge. Section 4 introduces our scheme, including the system model, security assumptions, private key distribution management of the multi-attribute mechanism, and construction of structured data & user attribute permission tree. Section 5 gives the encryption scheme. Section 6 is the security proof analysis and performance evaluation. Finally, a full-text summary is made in Section 7.

2 Related Works

Taking the medical application as an example, we used "TS = (access) AND TS = (blockchain) AND (TS = (medical) OR TS = (health))" to search the literature in the past five years, and obtained 789 and 871 records in WEB OF SCIENCE and SCOPUS databases, respectively.

2.1 Blockchain Used in Access Control

After removing repeated literature and summary analysis, we find that the combination of blockchain technology and access control forms two decentralized access management methods: one is the combination of blockchain technology and existing access control models, blockchain acts as the carrier of existing access control models, including the combination of RBAC [8], ABAC [9] and Cap-BAC [10]. The other is a newly developed access control method based on transactions [11] or intelligent contracts [12]. Although the combination of blockchain and access control can solve the problems of a single point of failure and lack of trust, it also introduces problems such as complex models. Consensus mechanism optimization, storage expansion, and privacy protection have become new challenges in access control for blockchain applications.

2.2 Attribute Revocation Based on CP-ABE

The above research uses many cryptographic tools, such as CP-ABE (For a detailed process, please refer to Section 3.3), proxy re-encryption, and zero-knowledge proof. Furthermore, we add the keywords "Policy" and "Attribute" to search again, and get 185 and 228 results in the above database, respectively. It was found that 156 articles are using the CP-ABE tool to solve access control problems based on blockchain, accounting for 21.3%. CP-ABE has become an important way to solve access control problems by taking advantage of its one-to-many sharing, but attribute revocation, ciphertext search, and policy hiding have not been well solved, which limits the application of the scheme.

Pirretti et al. [13] proposed the concept of attribute revocable for the first time. Since then, the research on attribute revocation has developed in two directions, one is user-level revocation, and the

other is attribute-level revocation. Because the latter is more flexible, it has become the current research hotspot. Wei et al. [3] proposed an attribute-based data access control method for cloud storage systems. All attribute institutions in the system independently issue keys for users, but this process only supports user-level revocation. Sowjanya et al. [14] proposed a secure WBAN framework, which uses CP-ABE based on elliptic curve cryptography without bilinear pairing operation. Yang et al. [4] designed a revocable CP-ABE blockchain multi-permission EHR sharing scheme (MA-RABE) using a distributed one-way anonymous key protocol to address the power centralization caused by single attribute permissions in the current CP-ABE scheme, as well as the curious or even malicious issues of cloud servers. Chaudhary et al. [15] proposed a multi-authorization CPABE scheme for Internet of Things devices, which eliminates the most expensive ciphertext update operations in the process of attribute revocation and addition for the first time. Considering the problems of the single-attribute authorization mechanism, Literature [16] proposed a blockchain-based secure and fair data trading system by taking advantage of the smart contract and matchmaking encryption. The proposed system enables bilateral authorization, where data trading between a seller and a buyer is accomplished only if their policies, required by each other, are satisfied simultaneously.

2.3 Searchable Ciphertext

In 2000, the scheme of searchable encryption appeared for the first time. Song et al. [17] used the cloud server's computing power to search the stored data's keywords to improve retrieval efficiency. In 2004, Boneh et al. [18] introduced searchable encryption technology into asymmetric encryption schemes for the first time. Since then, searchable encryption has developed into symmetric searchable encryption (SSE) and asymmetric searchable encryption (ASE). Literature [19] mainly realized two main features: avoiding the illegal sharing of private keys by multiple users, providing data access rights for data owners, and flexibly changing access policies. It adopts different traceable dynamic policies and provides a multi-access system with high security. Literature [20] proposed an attribute-based searchable encryption scheme, which supports the check function of multiple keywords. literature [21] implemented a secure search for ciphertext keywords in an open and transparent blockchain environment, and can resist chosen-plaintext attacks and keyword-guessing attacks., Wang et al. [22] proposed an effective lattice-based public-key searchable encryption, which is suitable for fine-grained access control of edge computing.

2.4 Policy Hiding Based on CP-ABE

The function of policy hiding is mainly realized through the transformation of access structure, which includes four kinds: Linear Secret Sharing Scheme (LSSS), "and gate" structure, Access Tree, and monotone Boolean formula (MBF). Belguith et al. [23] proposed a policy-hidden ABE scheme, which replaces user attributes in the access control structure with computable values. Tao et al. [24] used a complete hiding strategy, using a hash function to map the user's attribute values to random numbers. The CP-ABE of partial hiding strategy was proposed in the literature [25], but the revocation of user attributes cannot be realized. Literature [4,5] only needed to re-encrypt the ciphertext when revoking, but it can only realize the revocation of the user. Literature [26] provided blockchain-based data sharing using hidden vector encryption (HVE) with "conversion steps". Literature [27] implemented an efficient FHP-CP-ABE scheme using an Attribute Bloom Filter (ABF), but this scheme only supports And-gate-related access structures. Literature [28] proposed a hidden policy-based attribute encryption scheme (HP-CPABKS) with keyword search. Users with mismatched attributes will not be able to search encrypted data and obtain any information about the access

structure. Zhang. et al. [29] proposed an efficient and secure sharing scheme (PHAS-HEKR-CP-ABE) based on LSSS, but the complexity of their decryption testing algorithm increased exponentially. Table 2 shows the functional comparison of several schemes. In addition, for the medical image data involved in this article, the methods proposed in [29–31] can further protect the privacy of user image data.

Table 2: Comparison of solutions

Solutions	[2]	[4]	[23]	[24]	[25]	[5]	[26]	[27]	[28]	[29]
Multi-attribute	×	×	✓	✓	✓	×	×	×	×	×
Strategy hide	FHP	×	FHP	FHP	PHP	FHP	FHP	FHP	FHP	PHP
Revocation based on PRE	×	Att	×	User	User	User	×	×	×	×
Ciphertext updates	✓	✓	×	✓	✓	✓	×	×	×	×
Key update	✓	✓	×	✓	✓	×	×	×	×	×
Access structure	LSSS	LSSS	LSSS	LSSS	AND	AND	LSSS	AND	AND	LSSS
Outsourced	×	✓	✓	✓	×	✓	✓	✓	✓	×

3 Preliminary

To better understand the proposed program, in this section, we will provide the main technologies and necessary background for the program.

3.1 Blockchain Combines with Raft and IPFS

Blockchain [32] is essentially a distributed database maintained by all participants. The database uses cryptographic algorithms and timestamps to string all transactions in the network in the form of chains in time order to ensure that the data cannot be tampered with and forged, to achieve an environment without trust. The consensus algorithms of blockchain include POW, PBFT, DPOS, and so on. In the DPOS mechanism, N nodes are selected to participate in accounting, and PBFT reaches a consensus through a consensus agreement among the limited number of nodes in the alliance chain. Although the number of nodes is limited, it also requires pairwise communication between nodes to complete the consistency protocol. On the other hand, the Follower of Raft synchronizes the account books directly according to the instructions of the Leader, and there is only one accounting node in the same period. Raft [33,34] is direct and efficient and can achieve block output in seconds. Each node in a distributed cluster that follows the Raft algorithm plays one of the following three roles: The Leader, is responsible for communicating with the client, receiving commands from the client, and forwarding them to the follower to complete data synchronization; the Follower, its meticulous execution of commands from the leader; Candidate, when a follower has not heard from a leader for a long time, he will rise and become a candidate, grabbing the qualification to become a leader. In our scheme, multiple candidate AA will select the leader AA based on the Raft consensus mechanism to complete the distribution of the attribute private key.

A classic block structure is shown in Fig. 1, which consists of two parts: block head and block body. The blockhead mainly consists of the version number, the hash value of the previous header, the timestamp, the random value, and the Merkel root. Block is a specific transaction record; each transaction information is digitally signed and hashed layer by layer to construct a Merkle tree to ensure that the transaction cannot be forged.

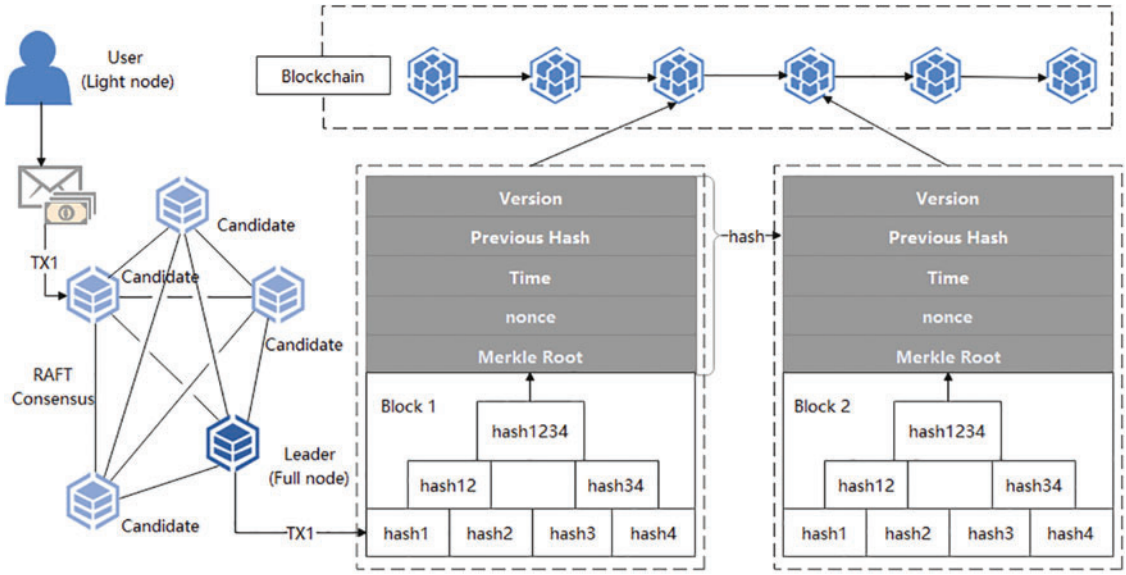


Figure 1: Blockchain structure and consensus

IPFS [35] technology solves the defect that the blockchain cannot store massive files. IPFS is based on content addressing, that is, what the user is looking for is not an address but the stored content, and does not need to verify the identity of the sender, but only needs to verify the hash of the content. Libp2p, a sub-module of IPFS, has been adopted by many blockchain projects, including Ethernet 2.0 and Polkadot, so this paper will use IPFS to store file data later.

3.2 Difficult Assumptions

3.2.1 BDH Assumption

Given a group $(g, g^a, g^b) \in G^3$, calculate function value $e(g,g)^{ab} \in G_T$. If there exists an algorithm $\mathcal{A}$ satisfying $\Pr\left[\mathcal{A}\left(g, g^a, g^b\right) = e\left(g,g\right)^{ab}\right] \geq \varepsilon$, claim that algorithm $\mathcal{A}$ can solve the BDH problem with an advantage of ε, where g is randomly from G, $a, b \in_R \mathbb{Z}_p$. We say that the BDH assumption in G holds if there is no PPT algorithm solving the BDH problem in G by a non-negligible advantage ε.

3.2.2 DBDH Assumption

Let (p, G, G_T, e) be a tuple of bilinear groups of order prime $p.g$ is the generating element of G. If given the challenge tuple $D = ((p, \mathrm{G}, \mathrm{G}_T, e), g, g^a, g^b, g^c)$ and Z. Then there is no PPT adversary $\mathcal{A}$ can distinguish $Z = Z_0 = e\left(g,g\right)^{abC}$ from $Z = Z_1 = e\left(g,g\right)^d$ by a non-negligible advantage ε, where the advantage of adversary $\mathcal{A}$ is: $Adv_{\mathcal{A}}^{DBDH}(\lambda) = |\Pr[\mathcal{A}(D, Z_0) = 0] - \Pr[\mathcal{A}(D, Z_1) = 0]|$, $a, b, c, d \in Z_p$.

3.2.3 HDH Assumption

Let h be an integer, and H: $\{0,1\}* \rightarrow \{0,1\}^h$ be a Hash function. Given $\left(g, g^a, g^b, H\left(g^c\right)\right) \in G^3 \times \{0,1\}^h$ as input, If $a \cdot b = c$, then output "1", otherwise output "0". An algorithm $\mathcal{A}$ solves the HDH problem by outputting $b' \in \{0,1\}$ with the advantage of ε, if it satisfies Eq. (1):

$$|\Pr[\mathcal{A}\left(g, g^a, g^b, H\left(g^{ab}\right)\right) = 1 : g \in_R G, a, b \in_R Z_p^*] - Pr[\mathcal{A}\left(g, g^a, g^b, \eta\right) = 1 : g \in_R \mathbb{G}, \eta \in_R \{0,1\}^h, a, b \in_R Z_p^*]| \geq \varepsilon \quad (1)$$

This paper claims that the HDH assumption holds if there is no PPT algorithm to solve the HDH problem with a non-negligible advantage ε.

3.3 Ciphertext Policy Attribute Encryption

The CP-ABE scheme consists of four algorithms: a system parameter building algorithm, a key generation algorithm, an encryption algorithm, and a decryption algorithm:

Step1. System parameter building algorithm (k): input security parameter k, output public key pk and master key mk.

Step2. Key generation algorithm (mk, Atr): enter the master key mk and the user attribute set Atr, and return the private key sk for the attribute set Atr.

Step3. Encryption algorithm (pk, M, A): input public key pk, message M and access structure A, output ciphertext CT.

Step4. Decryption algorithm (pk, C, sk): input public key pk, ciphertext C and private key sk about user attribute set Atr. If the attribute set can satisfy the access structure A of the ciphertext, the message M will be output, if not, the decryption will fail.

4 Proposed Scheme

4.1 System Model

The system model of our proposed scheme is shown in Fig. 2 and consists of seven entities: the set of Authorization Authorities (AAs), the data owner (DO), the data visitor (DA), the blockchain (BC), the proxy server (PS), the IPFS and the search server (SS).

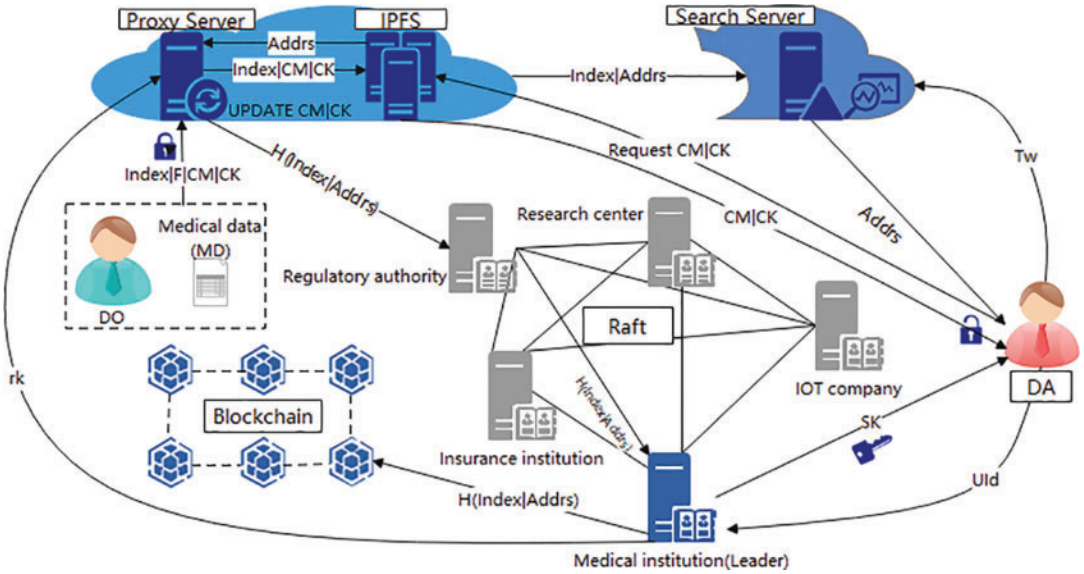


Figure 2: System model

The communication channels between the subjects are all secure, and users access the encrypted files in a read-only way and cannot modify them at will. The functions and security assumptions of each entity are described below.

AAs: AAs are composed of multiple attribute authorities that participate in attribute distribution, each representing a different organization, such as IoT companies, healthcare providers, insurance providers, healthcare regulators, etc. Each AA, as the whole node of the blockchain, runs the Raft consensus algorithm. Finally, the leader initializes the whole system distributes the attribute private key encrypted by the user's public key, and records the hash value of the IPFS storage location to form a block. In addition, the leader is also responsible for generating the proxy re-encryption key (rk) when the user attribute is revoked.

AA is semi-trusted, and the leader (supervising AA) will generate the system public key, master key, and user's private key, but may distribute attribute private key that does not match the user attributes. The supervising AA will supervise the distribution of private keys by the current AA. AA is always online and will not actively disclose the system master key and user attribute private key, and there is a method of securely transmitting the private key to the user.

DO: DO use a symmetric key (K) to encrypt the medical file (MD) to form data ciphertext (CM), defines access policy A, and encrypts K to get key ciphertext (CK) according to A. Then the generated index, CM, CK, K, conversion method F, and embedded access policy A are packaged and uploaded to the proxy server.

DA: All the users who want to access medical data are DA. DA generates keywords according to the demand, generates search trapdoor T_w, and then uses the public key (PK_{SS}) of the search server to encrypt and send it to the search server to obtain the ciphertext storage location (Addrs), and then DA applies for the attribute private key (sk_{DAi}) from the AA. For any DA, MD can be restored if and only if its attribute set satisfies the access policy A.

DA is untrusted, and collusion attacks may be launched if each cannot be decrypted independently.

BC: As a traceable distributed ledger, BC stores the records of attribute private key distribution and data verification in the system. Because the blockchain is publicly verifiable, AA hashes the storage location of CM and CK to form a data verification record.

PS: When the user attribute is revoked, PS uses the re-encryption key (rk) and F algorithm and re-encrypts CK and CM to prevent the PS from directly operating K and MD.

PS is semi-trusted. PS will faithfully execute users' requests for uploading ciphertext and re-encryption, but it will conspire with malicious users to steal sensitive information for commercial purposes.

IPFS: IPFS is honest and curious. IPFS honestly performs the operations requested by users, but it also attempts to obtain confidential data files and symmetric keys stored in it. In addition, IPFS is always online and provides a stable service.

SS: SS stores the ciphertext storage address (Addrs), and a keyword index, matches the trapdoor, and returns the Addrs to the user with a successful match.

SS is semi-trusted and does not expose the Addrs, but attempts to get the MD corresponding to the Addrs.

4.2 Security Model

The security model of this scheme is based on the IND-sAtt-CPA game, which simulates the attack game between Challenger and Adversary. In the game, the challenger simulates the execution of the algorithm to answer the query request of the Adversary. The specific process of the game is as follows:

Initialization phase: The adversary selects a challenge access policy and sends it to the challenger.

Build phase: The challenger runs the system parameter building algorithm to generate the public key and the master key (Pk, Mk) and sends the public key Pk to the attacker Adversary.

Phase 1: The adversary adaptively selects the set of attributes for the private key. The challenger runs a key generation algorithm to generate a private key for the attributes.

Challenge phase: The adversary sends two plaintexts of equal length m_1, m_2 to the challenger. Challenger chooses $b \in \{0, 1\}$ at random and returns The calculation process is shown in Eq. (2).

$$C_b = Encrypt(m_b, \tau^*, Pk) \tag{2}$$

Phase 2: The adversary repeats phase 1 and continues to Continue to ask for the private key.

Guessing phase: Adversary outputs $b' \in \{0, 1\}$, based on a guess.

Definition 1: A scheme is CPA-safe if the adversary's attack advantage in the IND-sAtt-CPA game is negligible in any probabilistic polynomial-time (PPT), where the advantage $\varepsilon = |Pr - 1/2|$.

4.3 Blockchain-Based Private Key Distribution Management

4.3.1 Raft Consensus Scheme Based on Credit

The Raft algorithm cannot select the only node as the leader when multiple nodes get the same number of votes, and the solution is to re-elect, which will increase the time and complexity of the system. In this paper, a credit score is used to improve the speed of Raft selection. If the node distributes the correct attribute private key, the node can increase the credit score, otherwise, reduces the credit score. After the leader is selected, a confirmation message is sent to other nodes, and the selection process ends, and the other nodes switch to the follower state and supervise the generation of new blocks, as shown in Fig. 3. The score is added based on the historical number of node attribute private key releases and the stability of the node, while the score is deducted based on the node's error feedback, penalty factors, etc. The credit score is calculated as shown in Eq. (3).

$$\begin{cases} R_i = aVs - \beta Vw \\ a = \dfrac{Ni}{N} \\ \beta = \begin{cases} \beta_1 & [0, R_1] \\ \vdots & \\ \beta_j & [R_{j-1}, R_j] \end{cases} \end{cases} \tag{3}$$

In Eq. (1), R_i is the credit score of the node i, Vs is the cumulative number of successful private key releases; Vw is the number of erroneous private key releases; a is the node stability factor; N is the number of days the system is running; Ni is the number of days that node i is not out of service, etc., β is the node penalty factor, which is the segmentation function of the current credit value; $\beta_1, \cdots, \beta_j$ are the segmentation function values corresponding to each credit value interval, generally $\beta_1 < \cdots < \beta_j$,

$R_1, \cdots, R_j$ are the corresponding interval boundary values. For more information on how to improve the efficiency of selecting leaders, see Section 6.4.3.

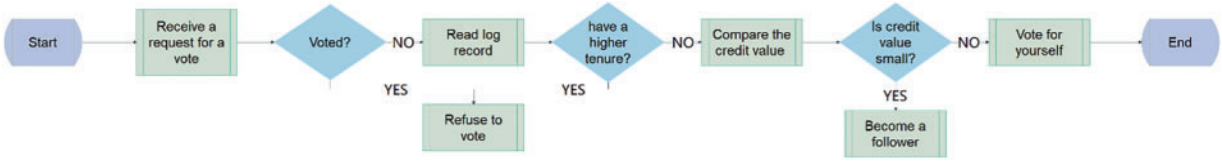


Figure 3: Voting process

4.3.2 Block Structure of Private Key Distribution

Whenever a user initiates an application for an attribute private key to AAS, all AA record its user number (UID) and attribute. Leader distributes the attribute private key encrypted based on the user blockchain public key and writes the data to TX when it is completed. TX includes message type (Type), timestamp (Time), UID, user attribute set (Attrset), and the hash value of SK and a random number (t). The leader also records H (Index | Addrs) submitted by other AA to ensure that the data is untampered and undeniable. After the leader completes this Block, the leader broadcasts it to the supervisor AA. Each supervisory AA randomly extracts the Block data, first uses its UID and Attrset tables to audit whether the Attrset in this TX is wrong, and then uses Attrset, t, PK, and MSK to calculate the hash value of SK, and audit whether there are errors in SK. If the audit fails, the ID of the block and the UID of the error will be broadcast. All AAs receiving the broadcast will review the data. If there is an error, the AA responsible for the data will be deducted a certain score. The leader will rewrite the data and broadcast the modified block. If it is determined that this block is legal, the AA responsible for this block will be awarded a score. At the end of his tenure, the leader packages all legal TXs, the specific structure of the block is shown in Fig. 4.

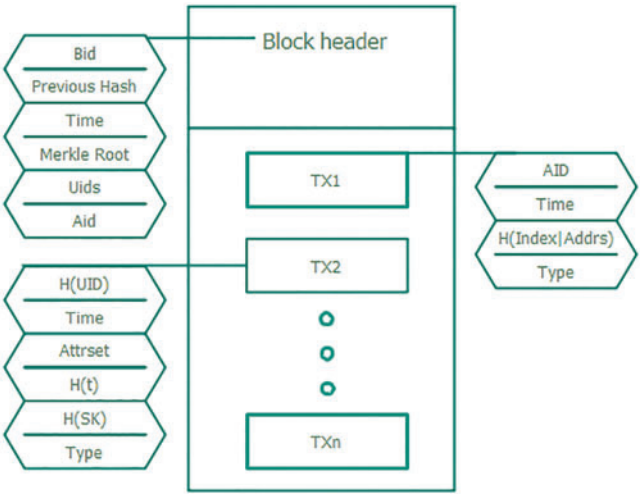


Figure 4: Block structure of private key distribution

When users find that there is a problem with their private key (their attribute set satisfies the access structure but cannot complete the decryption, or does not satisfy the access structure but can decrypt), they can appeal to AAS. Leader finds the corresponding block according to the time stamp of the appeal user's private key, then finds the user's private key distribution record according to the UIDs in the header of the Block, regenerates the private key, and rejects the appeal if there is no error. Otherwise, the credit score will be deducted from the AA responsible for the distribution of the user's private key, and the current leader will redistribute the user's private key.

4.4 Structured Data and Construction of User Attribute Permission Tree

In the introduction, this paper has introduced the structure of encrypted files, specifically, for example, the MD file in Fig. 5 contains ID and PD files, while the ID file can be divided into BP and SPO2H files. In other words, the access tree T-MD of the file MD is composed of T1-ID and T2-PD, while T1-ID is composed of T1-ID-SPO2H and T1-ID-BP, which has a structural relationship.

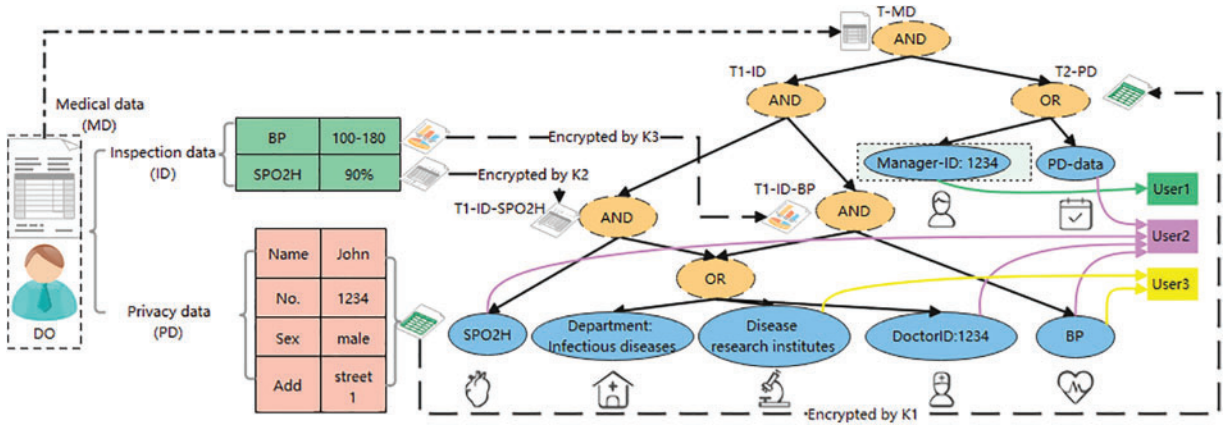


Figure 5: Structured data & user attribute permission tree

The existing CP-ABE needs to encrypt the attributes of each file separately, which is complicated. This paper designs a structured CP-ABE algorithm, which encrypts the files BP, SPO2H, and PD by using the symmetric key K (K1, K2, K3), while the symmetric key K only needs attribute encryption once.

For the above file hierarchy, the T-MD tree contains two leaf nodes, T1-ID and T2-PD. Access to T2-PD requires that the user has an attribute Manager-ID:1234 or an attribute PD-data. T1-PD also has two leaf nodes corresponding to files T1-ID-SPO2H and T1-ID-BP. To access T1-ID-SPO2H, the user needs to have the attribute SPO2H and one attribute in the Infective disease, research institutes, Doctor ID: 1234. To access T1-ID-BP, the user needs to have the attribute BP and one attribute in the Infective disease, research institutes, Doctor ID: 1234. The construction of the entire tree is based on AND and OR gates. Therefore, users with different attributes have access to different files.

For example, User1 (with attribute Manager-ID: 1234) obtains K1 and can only manage patient privacy information PD. Attending doctor User2 (with attributes PD-data, BP, SPO2H, DoctorID: 1234) obtains K1, K2, and K3 and can access all data, while research institution User3 (with attributes Disease research institutes, BP) can only obtain K3 and access BP data.

When AA wants to revoke or update the file access permission of a user, AA only needs to modify the user attributes, so that different users get different attribute private keys, and further obtain the symmetric key K (K1, K2, K3) of different files.

This paper assumes that there are files m1 and m2, and their access structure trees are T1, T2, T1, and T2 with hierarchical relationships. Our solution integrates them into a complete access tree. For m1 and m2, only one access policy-based encryption operation is required. If the hierarchical relationship of the files is not considered, separate attribute encryption operations need to be performed for m1 and m2. When the number of files is 2 for policy update, our encryption cost based on the file tree structure is O(2) $*$ symmetric encryption cost +O(1) $*$ asymmetric encryption cost, and the calculation complexity of the corresponding n files is O (n) $*$ symmetric encryption cost +O(1) $*$ asymmetric encryption cost, while the encryption cost of the non-file tree structure is o (n) $*$ asymmetric. Our solution significantly reduces computational complexity.

5 Encryption Scheme

There are two stages of system initialization: the user registration phase and the attribute authority initialization phase.

User registration: Each AA_i, DO, DA, PS, and SS are registered as a user of the blockchain, and generate their key pair (PK_{AAi}, SK_{AAi}), (PK_{DOi}, SK_{DOi}), (PK_{DAi}, SK_{DAi}), (PK_{PS}, SK_{PS}), (PK_{SS}, SK_{SS}), and distribute the identity A_{id} and U_{id} to each AA_i and user.

Attribute Authority initialization: This process is performed by the initial authority AA_i, as shown in Func1.

Func1: Setup $(k, A\ S)$

Input: System security parameter k and the set of all attributes in the system (AS)
Output: System public key (pk) and master key (mk)
Step1: Generate the bilinear group G and e: $G \times G \rightarrow G_T$,$G$ and G_T are cyclic groups of order N, Where $N = p * r$, p and r are different prime numbers. Use G_p and G_r to represent subgroups of order p and r of G, respectively. Use g_p and g_r represents the generators of G_p and G_r, respectively.
Step2: Generate random values $\alpha, t_1, t_2, \cdots, t_n \in Z_p^*, n$ is an integer, $R_1, R_2, \cdots, R_n \in G_r$, calculate $x = g_p \cdot R_1$, y = e $\left(g_p, g_r\right)^a$, $T_j = g_p{}^{t_j} \cdot R_j$ $(1 \leq j \leq n)$.
Step3: Output $pk = \left(e, x, y, T_j\right)$ *and* $mk = (\alpha, t_j)$.
Step4: Initialize version number $seq = 1$, AA save (seq, mk), publish (seq, pk).

Attribute Key generation phase. The process is executed by the initial AA_i, as shown in Func2.

Func2: SkGen (mk, pk, A)

Input: System public key (pk), master key (mk) and, user attribute set $U_{id}A$
Output: User Attribute Key sk_{DAi}
Step1: Select random $r \in Z_p^*$ Then calculate $d_0 = g_p{}^{a-r}$ $\forall a_j \in U_{id}A$, calculate $d_j = g_p{}^{rt_j^{-1}}$.
Step2: Calculate and output $sk_{DAi} = (seq, d_0, d_j)$, seq is the current version number.

Cryptographic generation phase. This process is executed by DO, as shown in Func3.

Func3: Enc (K, A, pk)

Input: The system public key (pk), the symmetric key K ($K1$, $K2$, $K3$) for the encrypted MD and the access structure A

Output: encryption result CK of K

Step1: Select random $s \in Z_p^*$, $R_0^{'} \in G_r$, then calculate $C_0 = x^s \cdot R_0^{'}, C_1 = m \cdot y^s = K \cdot e\left(g_p, g_p\right)^{as}$.

Step2: Assign secret s based on the access policy tree, Set the value of the root node to s, Shared secret value $s_i = f(i)$, where i denotes the location index of the leaf node and the value of each leaf node is used to generate a portion of the ciphertext. $f(i) = \sum_{j=0}^{t-1} b_j x^j$, where b_j is a random value and t is the number of child nodes.

Step3: For each leaf node, calculate $\forall a_{j,i} \in A$, $C_{j,i} = T_j^{s_i} R_j^{'}$, where $R_j^{'}$ is a random value, $R_j^{'} \in G_r$, $CK = (seq, C_0, C_1, \forall a_{j,i} \in A{:}[i, C_{j,i}])$.

Index generation phase. This process is executed by DO, as shown in Func4.

Func4: Index (pk, PK_{SS}, w)

Input: Search server's public key PK_{SS}, the system public key pk, keyword w

Output: $Index_w$

Step1: Select random value $\delta \in Z_p^*$. Then calculate $Index_w = \left[g^{\delta}, e\left(PK_{SS}, H_1(w)^{\delta}\right)\right]$.

Trap gate generation phase. This process is run by user DA, as shown in Func5.

Func5: Trapdoor (pk, PK_{SS}, w)

Input: Search server's public key PK_{SS}, the system public key pk, keyword w

Output: Trapdoor of search T_w

Step1: Select random value $\delta' \in Z_p^*$, then calculate $T_w = [g^{\delta'}, H_1(w) \cdot H((PK_{SS})^{\delta'})]$.

Search matching phase. This process is run by SS, as shown in Func6.

Func6: Compare (pk, $Index_w$, T_w, SK_{SS})

Input: The system public key (pk), $Index_w$, T_w, $SK_{SS} = u_{ss}$

Output: Storage addresses of CK and CM Addrs

Step1: Calculate $\partial = H_1(w) \cdot H((PK_{SS})^{\delta'} / H(g^{\delta' u_{SS}})$ if $e\left(PK_{SS}, H_1(w)^{\delta}\right) = e\left(g^{\delta}, \partial^{u_{SS}}\right)$.

Step2: If the keyword index $Index_w$ matches successfully with the T_w then output Addrs.

Decryption phase. The process is run by the user DA, as shown in Func7.

Func7: Decrypt (CK, sk_{DAi})

Input: Ciphertext CK And user attribute key sk_{DAi}

Output: Symmetric key K and MD

Step1: After obtaining the ciphertext CK, check whether the version number of CK is consistent with the version number of the sk_{DAi}, if not, visit the AA_i to update the private key.

Step2: If the version number is the same, calculate $K = C_1 / e(C_0, d_0) \prod_{a_j \in w} e\left(C_{j,i}, d_j\right)^{l_{i(0)}}$, where $l_{i(0)}$ is the Lagrangian factor.

(Continued)

Func7 (continued)

Step3: The user obtains data MD by decrypting the corresponding ciphertext CK according to the obtained symmetric key K.

Attribute revocation consists of four phases: the CK&CM ciphertext conversion phase, the proxy re-encryption key generation phase, the attribute private key update phase, and the proxy re-encryption phase. The process is in Func8.

Func8: Trans (CK, CM)

Input: Ciphertext CK, CM
Output: CK', CM'
Step1: Select random values $\varphi, k_1, k_2, k_3, \in Z_p^*$, take $K1, K2, K3$ as g^{k1}, g^{k2}, g^{k3}c alculate $c1' = g^{\varphi}$ $[K \cdot e\left(g_p, g_p\right)^{as}] = e\left(g_p, g_p\right)^{as} (g^{k_1+\varphi} | g^{k_2+\varphi} | g^{k_3+\varphi})$, $CK' = (seq{+}1, C_0, C_1', \forall a_{j,i} \in A{:}[i, C_{j,i}])$ Complete the update of symmetric key K.
Step2: Let m_1, m_2, m_3 to be the sub files of MD respectively, calculate g^{φ} $[g^{k1} m_1 \left| g^{k2} m_2 \right| g^{k3} m_3] = (g^{\varphi+k_1}) m_1 \left| (g^{\varphi+k_2}) m_2 \right| (g^{\varphi+k_3}) m_3$, complete the update to CM.

Proxy re-encryption key generation phase: this process is run by AA_i, as shown in Func9.

Func9: RkGen (l, mk)

Input: Collection of attributes to be updated $l \in A$ S, mk
Output: Proxy re-encryption key rk
Step1: For $\forall a_j \in l$, randomly choose $t_j' \in l$, calculate $rk_j = t_j'/t_j$.
Step2: Output $rk = (seq + 1, rk_j)$.

Attribute private key update phase: this process is run by AA_i, as shown in Func10.

Func10: SkUpdate (sk_{DAi}, rk, l)

Input: sk_{DAi}, rk, User's new property collection l
Output: User's new attribute key sk'_{DAi}
Step1: Compare whether the version number of sk_{DAi} is the same as that of rk if not, output $sk'_{DAi} = sk_{DAi}$.
Step2: If the version number is the same, then $\forall a_j \in l$, calculate $d_j' = d_j^{rk_j^{-1}}$.
Step3: Output $sk'_{DAi} = (\mathrm{seq} + 1, \mathrm{d}_0, \mathrm{d}'_j)$.

Proxy re-encryption phase: this process is run by PS, as shown in Func11.

Func11: ReEnc (CK, CK', rk, β)

Input: Converted symmetric key cipher CK', rk, Attribute Set β in Access structure
Output: Key ciphertext after proxy re-encryption CK''
Step1: Compare whether the version number of CK' is the same as that ofrkif not, output CK.
Step2: If the version number is the same, then $\forall a_j \in \beta$, $\beta \subseteq A$ S, calculate $C'_{j,i} = C_{j,i}^{\ rk_j}$.
Step3: Output $CK'' = (seq + 1, C_0, C_1, \forall a_{j,i} \in A{:}[i, C'_{j,i}])$.

The whole system operation flow is shown in Fig. 6.

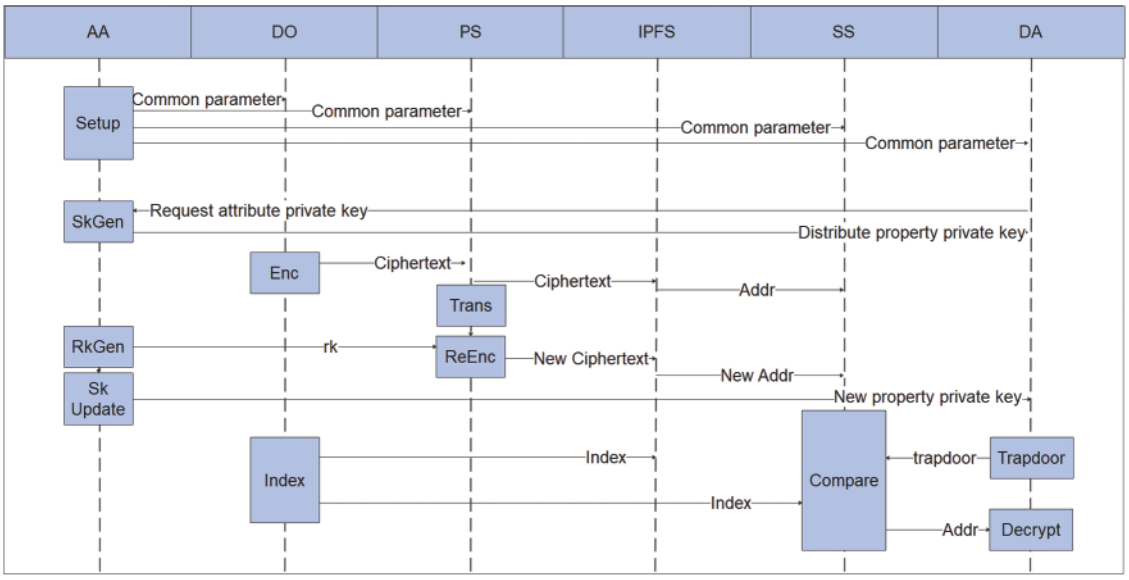


Figure 6: System operation flow

6 Proof and Evaluation

6.1 Proof of Correctness

6.1.1 Correctness of Decryption

The set of attributes of the user is noted as $U_{id}A$, the user's private key is sk_{DAi}, the access policy is noted as A. If the user's attribute set $U_{id}A$ does not satisfy the access policy A, the plaintext will not be decrypted according to the decryption algorithm, otherwise, the plaintext K can be decrypted. The decryption process is as follows:

$$\begin{aligned}\hat{K} &= \frac{C_1}{e\left(C_0, d_0\right) \prod_{a_j \in U_{id}A} e\left(C_{j,i}, d_j\right)^{l_{i(0)}}} \\ &= K \cdot \frac{e\left(g_p, g_p\right)^{as}}{e\left(g_p^s, g_p^{a-r}\right) e\left(R_0^s R_0', g_p^{a-r}\right)} \cdot \frac{1}{\prod_{a_j \in U_{id}A} e\left(g_p^{t_j \cdot s_i}, g_p^{rt_j^{-1}}\right)^{l_{i(0)}} \cdot e\left(R_j^{s_i} R_j', g_p^{rt_j^{-1}}\right)^{l_{i(0)}}} \\ &= K \cdot \frac{e\left(g_p, g_p\right)^{as}}{e\left(g_p^s, g_p^{a-r}\right) e\left(g_p, g_p\right)^{rs}} = K\end{aligned} \tag{4}$$

When attribute revocation occurs, there are:

$$rk_j = t_j'/t_j, d_j' = g_p^{rt_j^{-1} \cdot rk_j^{-1}}, C_{j,i}' = C_{j,i}^{rk_j} = (T_j^{s_i} \cdot R_j')^{rk_j} = g_p^{t_j \cdot s_i \cdot rk_j} \cdot R_j^{s_i}, \tag{5}$$

Then

$$\prod_{a_j\in\beta} e\left(C'_{j,i}, d'_j\right)^{l_{i(0)}} = \prod_{a_j\in\beta} e\left(g_p^{t_j\cdot s_i\cdot rk_j}\cdot R_j^{s_i\cdot rk_j}, g_p^{r\cdot t_j^{-1}\cdot rk_j^{-1}}\right)^{l_{i(0)}} = e\left(g_p, g_p\right)^{rs}, \tag{6}$$

Then

$$\frac{C_1}{e\left(C_0, d_0\right)\prod_{a_j\in\beta} e\left(C'_{j,i}, d_j\right)^{l_{i(0)}}} = K\cdot\frac{e\left(g_p, g_p\right)^{as}}{e\left(g_p^s, g_p^{a-r}\right)e\left(g_p, g_p\right)^{rs}} = K, \tag{7}$$

Proof complete.

6.1.2 Correctness of Matching Search Trapdoors

For $\forall\delta, \delta' \in Z_p^*$, let $A = g^{\delta}$, $B = e(PK_{SS}, H_1(w)^{\delta})$ denote the index corresponding to the ciphertext containing the keyword w. $T_{w'} = [T'_1, T'_2]$ denotes the trapdoor corresponding to the keyword w', where $T'_1 = g^{\delta'}$, $T'_2 = H_1(w')\cdot H((PK_{SS})^{\delta'})$.

SS Calculate

$$\tau' = \frac{T'_2}{H\left((T'_1)^{u_{ss}}\right)} = H_1(w')\cdot\left(\frac{H\left(PK_{SS}\right)^{\delta'}}{H\left((T'_1)^{u_{ss}}\right)}\right) = H_1(w')\cdot\left(\frac{H\left(g^{u_{ss}}\right)^{\delta'}}{H\left(\left(g^{\delta'}\right)^{u_{ss}}\right)}\right) = H_1(w'), \tag{8}$$

Then

$$e(A, \tau'^{u_{ss}}) = e(g^{\delta}, (H_1(w'))^{u_{ss}}) = e(g, H_1(w'))^{\delta\cdot u_{ss}}, \tag{9}$$

And

$$B = e(PK_{SS}, H_1(w)^{\delta} = e(g^{u_{ss}}, H_1(w))^{\delta} = e(g, H_1(w))^{u_{ss}\cdot\delta}, \tag{10}$$

Then when the keyword $w' = w$, there is $e(A, \tau'^{u_{ss}}) = B$, i.e., index matches successfully with $T_{w'}$.

Proof complete.

6.2 Proof of Security

1) Theorem 1: If the decisional DBDH assumption holds. Then any PPT adversaries cannot selectively break our scheme.

Proof: Suppose there is a PPT attacker $\mathcal{A}$ with a non-negligible advantage ε against our IND-sAtt-CPA scheme. This scheme will build a PPT challenger $\mathcal{B}$ to simulate our scheme. The challenger $\mathcal{B}$ first sets the order of the bilinear group G to $N = p\times r$, $e\colon G\times G\to G_T$, where p and r are different prime numbers, G and G_T are cyclic groups. Denote the subgroups of order p and r of G by $\mathrm{G_p}$ and $\mathrm{G_r}$, respectively. $\mathcal{B}$ choose $\mathrm{u} =_R \{0,1\}$, when $\mathrm{u} = 0$, $Z_u = e(g_p, g_p)^{\theta}$, when $\mathrm{u} = 1$, $Z_u = e(g_p, g_p)^{abc}$. Challenger $\mathcal{B}$ constructs DBDH challenge $\left(g_p, A, B, C, Z_u\right) = (g_p, g_p^a, g_p^b, g_p^c, Z_u)$. The game between $\mathcal{B}$ and $\mathcal{A}$ is as follows.

Init: $\mathcal{A}$ chooses an access policy τ', and sends it to the challenger $\mathcal{B}$.

Setup: $\mathcal{B}$ choose random $x' \in Z_p^*$, Set $\alpha = ab + x'$, calculate $y = e(g_p, g_p)^{\alpha} = e(g_p, g_p)^{ab} e(g_p, g_p)^{x'}$, randomly selected $t_j \in Z_p^*$, $R_j, R_0 \in G_r (1 \le j \le n)$, and calculate

$$\forall a_j \in U_{id} A: T_j = \begin{cases} g_p^{b/t_j} \cdot R_j, a_j \notin \tau^* \\ g_p^{t_j} \cdot R_j, a_j \in \tau^* \end{cases} \tag{11}$$

$\mathcal{B}$ then sends the following public key $pk = (g_p * R_0, T_j, (1 \le j \le n))$ to attacker $\mathcal{A}$.

Phase1: $\mathcal{A}$ Select attribute set $\omega_j = \{a_j | a_j, a_j \notin \tau^*\}$. Then send the private key query request to the user, for each query request from $\mathcal{A}$, $\mathcal{B}$ chooses a random value $\forall r' \in Z_p$, and set $r = ab + r'b$. Then: $d_0 = g_p^{a-(ab+r'b)} = g_p^{x'}(g_p^b) - r'$. Since $a_j \notin \tau^*$, then $d_j = g_p^{rt_j/b} = \left(g_p^a\right)^{t_j} g_p^{r't_j}$. $\mathcal{B}$ sends user private key $sk_{DAi} = \left(d_0, d_j\right)$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two plaintexts m_0, m_1 to $\mathcal{B}$, $\mathcal{B}$ randomly choose a random m_b from two, $b \in \{0, 1\}$. $\mathcal{B}$ encrypts the message $C_0 = g_p^c \cdot R_0^c \cdot R_0^{'}$, $C_1 = m_b \cdot e\left(g_p, g_p\right)^{abc} e\left(g_p^c, g_p^{x'}\right)$.

Phase2: $\mathcal{A}$ repeat the first phase of the private key inquiry.

Output: Attacker $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$, if $b' = b$, the challenger can guess $u = 1, Z_u = e(g_p, g_p)^{abc}$, Because $Z_u = e(g_p, g_p)^{abc}$ is a valid ciphertext in the system, with the help of attacker $\mathcal{A}$, Challenger $\mathcal{B}$ can solve the DBDH problem with the following advantages: $Pr[b' = b | Z_u = e(g_p, g_p)^{abc}] = 1/2 + \varepsilon$, otherwise $\mathcal{B}$ guesses that $u = 0, Z_u = e(g_p, g_p)^{\theta}$ is a random ciphertext of A. The attacker cannot obtain any information about the plaintext message m_b. Therefore, Challenger $\mathcal{B}$ solves the DBDH problem with the following advantages: $\Pr[b' \ne b | Z_u = e(g_p, g_p)^{\theta}] = 1/2$. In summary, regardless of the conjecture, B solves the DBDH problem with the following advantage: $1/2Pr[u' = u | u = 0] + 1/2Pr[u' = u | u = 1] - 1/2 = \varepsilon/2$.

However, there is no effective polynomial algorithm to solve the DBDH problem with non-negligible advantages. Therefore, the attacker can not win the IND-sAtt-CPA game with the advantage of exceeding ε, and the attacker has no advantage in breaking into the system, that is to say, the attacker can not obtain the real attributes of the user through the ciphertext guess of the attribute key, so the encryption scheme prevents the disclosure of user attributes.

Proof complete.

2) Theorem 2: If the HDH assumption holds, then our proposed scheme satisfies the security against selective keyword attacks under the indistinguishability of trapdoors.

Proof: Assume that adversary $\mathcal{A}$ can attack our proposed solution by advantage of ε and makes at most q trapdoor queries, where $q > 0$. This scheme constructs a challenger $\mathcal{B}$ to solve the HDH problem in G with the advantage of ε. $\mathcal{B}$ input a random tuple of HDH challenge $\left(g, g^c, g^d, \eta\right) \in G^4$ and $H: \{0, 1\} * \rightarrow G$, Where H is a hash function and η is in the form of $H(g^{cd})$, or a random element in G. $\mathcal{B}$ randomly select $\delta \in Z_p^*$, let $h = (g^c)^{\delta}$. At the same time, $\mathcal{B}$ randomly chooses the hash function H_1. The game between $\mathcal{B}$ and $\mathcal{A}$ is as follows.

Setup: $\mathcal{B}$ randomly select $l \in Z_p^*$, Let the public key of SS be $PK_{SS} = (PK_{S1,} PK_{S2}) = ((g^c)^l, h^{1/cl}) = ((g^c)^l, h^{\delta/l})$. Then there exists an unknown c such that $SK_{SS} = u_s = cl$, and $e\left(PK_{S1,} PK_{S2}\right) = e(g, h)$.

Trapdoor1: When adversary $\mathcal{A}$ enquires the keyword w_j, Challenger $\mathcal{B}$ randomly selects $\delta' \in Z_p^*$, calculate $T_1^* = g^{\delta'}$, $T_2^* = H_1(w') \cdot H((g^c)^{l\delta'})$. $\mathcal{B}$ answers to $\mathcal{A}$ The trap door of w_j is $T_{wj} = [T_1, T_2]$. Where $l \in Z_p^*$ is the value selected at the system establishment stage.

Challenge: Adversary $\mathcal{A}$ outputs two keywords that will be challenged, w_1 and w_2, and Challenger $\mathcal{B}$ generates a challenge trapdoor $T_{wb}{}^* = [T_1{}^*, T_2{}^*]$, $T_1{}^* = (g^d)^{1/l}$, $T_2{}^* = H_1(w_{\tilde{b}}) \cdot \eta$, η is a challenge element in the HDH problem.

If $\eta = H(g^{cd})$, then $T_{w\tilde{b}}{}^*$ is a valid form of the trapdoor for the keyword $w\tilde{b}$ and satisfies a uniform distribution. If $g^c, g^d, H(\eta)$ is uniformly distributed in G, then, from the point of view of the attacker $\mathcal{A}$, $T_{w\tilde{b}}{}^*$ and $\tilde{b}$ are independent of each other.

Trapdoor2: $\mathcal{A}$ asks a trapdoor question about the keyword w_j, but $w_j \neq w_1, w_2$. $\mathcal{B}$'s answer is like the previous one.

Output: Finally, $\mathcal{A}$ outputs guess $b' \in \{0,1\}$, If $b = b'$, $\mathcal{B}$ outputs "1", i.e., $\eta = H(g^{cd})$), and $\left|Pr\left[\tilde{b} = b'\right] - 1/2\right| > \varepsilon$. Otherwise, $\mathcal{B}$ outputs "0", i.e., $\eta \neq H(g^{cd})$, and $|Pr[b = b']| = 1/2$. This leads to $|Pr[\mathcal{B}\left(g, g^c, g^d, H\left(g^{cd}\right)\right) = 0] - Pr\left[\mathcal{B}\left(g, g^c, g^d, \eta'\right) = 0\right]| \geq |(1/2 \mp \varepsilon) - 1/2| = \varepsilon$. In the above simulation, the probability of Challenger $\mathcal{B}$ terminating the game during the trapdoor questioning phase and the challenge phase is 0. Therefore, the probability of Challenger $\mathcal{B}$ eventually succeeding throughout the game is the probability of adversary $\mathcal{A}$ breaching the scenario.

For the bit-based plaintext encryption scheme, the adversary $\mathcal{A}$ cannot correctly obtain the corresponding bit plaintext under the condition of obtaining the challenge ciphertext, that is, the probability of the adversary's successful attack is 1/2. Then the encryption scheme is IND-CPA secure. The adversary cannot obtain any information leaks corresponding to the keyword search trap.

Proof complete.

6.3 Functional Comparison

Compare the functionality of the different options, as shown in Table 3. Only literature [36] and this scheme achieves multi-attribute authorization, which can prevent collusion attacks, but literature [36] is less flexible and cannot achieve user/attribute revocation. The scheme of [6] realizes user revocation and outsourcing decryption, but this scheme cannot prevent collusion attacks and cannot realize the public verifiability of private key distribution. Although both literature [7] and this paper use a symmetric key to encrypt files, only this paper realizes the update of the symmetric key and its ciphertext and revocation in a complete sense. In addition, literature [2,37] only implemented the function of policy hiding. Our proposed scheme also supports searchable encryption and introduces blockchain to prevent repudiation. The most important thing is that our scheme can manage rights in a complete sense in terms of users, user attributes, and file attributes.

Table 3: Functional comparison

Solutions	Access structure	Strategy hide	Multi-attribute	Revocation			Symmetric key update	Collusion resistance	Traceability	Searchable
				User	Attribute	File				
[6]	LSSS	✓	×	✓	✓	×	N/A	×	×	×
[36]	N/A	×	✓	×	×	×	N/A	✓	×	×
[2]	LSSS	✓	×	×	×	×	N/A	×	×	×
[37]	LSSS	✓	×	×	×	×	N/A	×	×	×
[7]	Access ree	×	×	✓	✓	×	×	×	×	×
Ours	Access ree	✓	✓	✓	✓	✓	✓	✓	✓	✓

6.4 Evaluation

6.4.1 Complexity Evaluation of our CPABE Scheme

In terms of performance, this paper compares the proposed scheme with [2,6,7,36,37] in storage overhead and computational overhead from a theoretical perspective.

To further illustrate the advantages of the proposed scheme in computational overhead, this paper refers to [2]. Based on the Charm open-source library, this scheme selects the elliptic curve group of 160-bit on the singular curve $y^2 = x^3 + x$ on the machine of AMD-R5-4500U, 2.30 GHz with 8.0 GB RAM, and get $T^p = 3.589\,ms$, $T^e_G = 2.821\,ms$, $T^e_{G_T} = 0.566\,ms$. This scheme assumes that the total number of attributes of the system is 20. The number of symmetric keys is 5. The experimental results of the time cost are shown in Fig. 7.

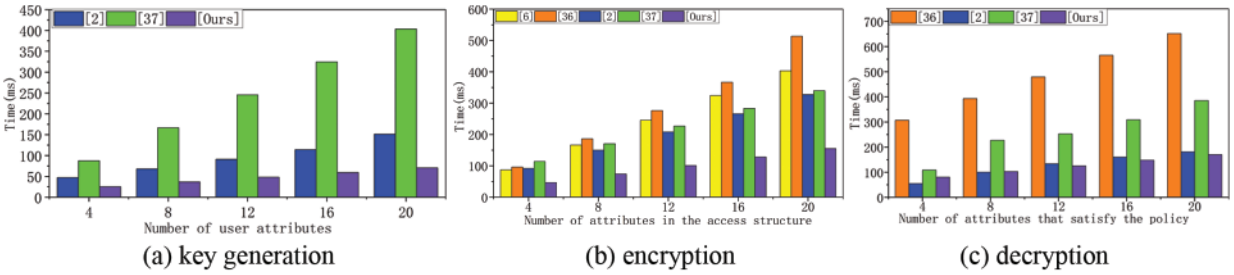


Figure 7: Comparison of time overheads

In the storage overhead comparison, this scheme only compares data owners and data visitors, because other entities do not have storage resource constraints, and to make the comparison more targeted, For ease of representation, Table 4 lists the meaning of each symbol.

Table 4: Meanings of notations

Notations	Meaning
$\lvert Z_p^* \rvert / \lvert G \rvert / \lvert G_T \rvert$	Size of element in $\lvert Z_p^* \rvert / G / G_T$
n	Number of system attributes
$\lvert A_u \rvert$	Number of leaf nodes in the access structure
$\lvert A_c \rvert$	Number of attributes in the access structure
$\lvert A_d \rvert$	Number of attributes of a user
N_K	Number of symmetric keys
l	Number of random numbers in G
$T^e_G / T^e_{G_T}$	Time for a group exponential operation in G / G_T
T^p	Time for a pairing operation $e\colon G \times G \rightarrow G_T$
mG	A multiplication operation in G
mG_T	A multiplication operation in G_T

As shown in Table 5, the storage overhead of [36] is the largest. In our scheme, the public key, master key, attribute private key and ciphertext are all the smallest, so the storage overhead is the least. In addition, for the communication cost, although the ciphertext occupancy space in our scheme is limited compared with the [7], our scheme achieves a complete sense of privilege management.

Computational complexity is shown in Table 6. In the scheme of [6,36], there is no computational overhead in the key generation phase because the multi-agency model suffers additional overhead. Compared with other [2] and [37], the main increased complexity of this scheme lies in the introduction of the symmetric key, which is similar to the consumption of [7]. In terms of attribute revocation, references [2,36,37] did not implement attribute revocation. In comparison, reference [6] did not use PRE to implement attribute revocation, which is the highest cost. Reference [7] and our scheme are both based on PRE implementation, and This paper has achieved full functionality revocation, including symmetric keys, attribute private keys, and ciphertext of symmetric keys, with the lowest computational cost. Func8 in PRE ensures that the proxy server can update the ciphertext of the symmetric key without obtaining the symmetric key plaintext, preventing the proxy server from snooping on users' privacy.

Table 5: Comparison of communication and storage overhead

Solutions	Public key	Master key	private key	Ciphertext
[38]	$5\|G\| + \|G_T\|$	$3\left\|Z_p^*\right\|$	$(2\|A_d\| + 3)\|G\|$	$(4\|A_c\| + 5)\|G\| + \|G_T\|$
[36]	$5\|G\| + \|G_T\|$	$(l + 5)\|G\|$	$(\|A_d\| + 9\mathrm{n})\|G\|$	$(4\|A_c\| + 4n)\|G\| + \|G_T\|$
[2]	$7\|G\| + \|G_T\|$	$\|G\| + 5\left\|Z_p^*\right\|$	$(2\mathrm{n} + 4)\|G\|$	$(3\|A_c\| + 4)\|G\| + \|G_T\|$
[37]	$7\|G\| + \|G_T\|$	$\|G\| + 2\left\|Z_p^*\right\|$	$(4\|A_d\| + 2)\|G\|$	$(5\|A_c\| + 2)\|G\| + 2\|G_T\|$
[7]	$3\|G\| + \|G_T\|$	$\|G\| + \left\|Z_p^*\right\|$	$(2\|A_u\| + 1)\|G\|$	$(\|A_c\| + 3)\|G\| + \|G_T\|$
Ours	$2\|G\| + \|G_T\|$	$2\left\|Z_p^*\right\|$	$(\|A_u\| + 1)\|G\|$	$(\|A_c\| + 1)\|G\| + G_T\|$

Table 6: Comparison of computational complexity

Solutions	Key generation	Encrypt	Decryption	Revocation	Revocation level	Revocation method	Fully updated
[6]	O(1)	$(7\|A_c\| + 3)\ T_G^e$	O(1)	$(8 + 3\mathrm{n} + 5\|A_c\|)$ $T_G^e + T^p + (1 +$ $2n + 2\|A_c\|)$ mG	Attribute	Other	✓
[36]	O(1)	$(2\left\|A_d\right\| + 5\|A_c\|$ $+2)T_G^e$	$(3\mathrm{n} + 6\|A_c\|)\ T^p +$ $2T_G^e$	×	×	N	×
[2]	$(2\|A_d\| + 4)\ T_G^e$	$(5n + 5\|A_c\|)\ T_G^e$	$\leq \left((2\|A_c\| + 1)\ T^p\right.$ $\left.+\|A_c\|\ T_{G_T}^e\right)$	×	×	N	×
[37]	$(7\|A_d\| + 3)\ T_G^e$	$(8n + 5\|A_c\|)\ T_G^e +$ $2T_{G_T}^e$	$\leq \left((5\|A_c\| + 1)\ T^p\right.$ $\left.+2\|A_c\|\ T_{G_T}^e\right)$	×	×	N	×
[7]	$\left(\left\|A_d\right\| + k\right)\ T_G^e$	$(2\|A_c\| + N_K)\ T_G^e +$ $2\left(\|A_u\| + k\right)\ T_{G_T}^e$	$5T^p + (\|A_c\| +$ $2N_K + 4)T_G^e$	$(2\|A_u\| + 2)$ $T_G^e + 2\|A_c\|\ T_{G_T}^e)$	Attribute	PRE	×
Ours	$\left(\left\|A_d\right\| + k\right)\ T_G^e$	$(2\|A_c\| + N_K)\ T_G^e +$ $2\left(\|A_u\| + k\right)\ T_{G_T}^e$	$5T^p + (\|A_c\| +$ $2N_K + 4)T_G^e$	$(\|A_u\| + 5)T_G^e + T^p$	Attribute	PRE	✓

Fig. 7a shows that the user attribute key generation time varies with the size of the user attribute set, and the calculation time increases with the increase of the user attribute set. Since the calculation time in [6,36] is constant, no comparison is made here. In the case of the same user attribute set size, the computing time of our proposed scheme is the least compared with the other two schemes, and the computing time of [37] is the largest. Fig. 7b shows that the encryption time varies with the number of attributes in the ciphertext, and the computing time increases with the increase of the number of attributes in the embedded ciphertext. In the case of the same number of embedded ciphertext attributes, the computing time of our proposed scheme is the least, while that [36] has the

largest calculation time. Fig. 7c shows that the decryption time varies with the minimum attribute set size that satisfies the access policy. Reference [6] can outsource the bilinear pairing operation with high computational overhead to the cloud computing platform, so the decryption computing time is constant and will not be compared here. The decryption computation time of other schemes increases with the increase of the minimum attribute set satisfying the access policy. Although the efficiency of our scheme is like that of the scheme [2], this paper has implemented the property revocation function.

6.4.2 Load Evaluation of Blockchain and IPFS

The additional load of the blockchain system is shown separately in Fig. 8. Considering the additional storage overhead caused by the blockchain, assume that the AA's number is 10, and the audit of the data in the Block by the supervisory is 2%. A block packages 100 private key distribution records and 100 storage addresses. When the number of authorities is 10, the number of system attributes is 12, and the head size is ignored, the block size is only 7.12 MB. Fig. 8 shows the additional storage overhead and time overhead as the number of private keys generated increases in the blockchain. In the current environment where storage media are getting cheaper and cheaper, it is entirely acceptable to exchange space and time for security.

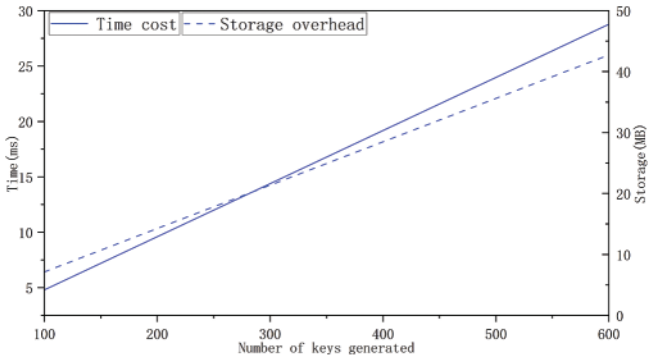


Figure 8: The overhead caused by blockchain

In the current work, the patient's detailed data will be stored in IPFS, while the blockchain mainly records the distribution of attribute keys and the storage address of patient information in IPFS.

In Fig. 8, this paper takes the transaction rate of the FISCO BCOS blockchain as a reference and calculates that the time cost caused by the block is not large. If the system is built on the bottom of BCOS, the peak value of the system can reach 20000TPS. If the sensor data is submitted 3 times per second for calculation, it can provide services for nearly 7000 people per second, so it is practical on a certain scale.

In addition, a large number of data reads and writes mainly occur in IPFS, because IPFS stores all case and sensor information, this paper tested the performance of IPFS (Table 7) on the local area network and ALIYUN, respectively, and each group carried out 5 repeated tests, taking the average as the test results.

Take the I7-8700CPU and ubuntu18.04-LTS systems with the main frequency of 3.7 Ghz as an example, for 1m files, the write rates in the local area network and Aliyun are 7.35 Mbit/s and 9.09 Mbit/s, respectively. Taking the single data transmission volume of the sensor as 1 kbit, it can complete the transmission of about 9000 sensor data per second. When each person uses three kinds

of sensors, it can provide services for 3000 people per second. Therefore, whether in terms of the efficiency of blockchain or IPFS, the proposed model can meet the use of a certain scale.

Table 7: IPFS read/write performance test

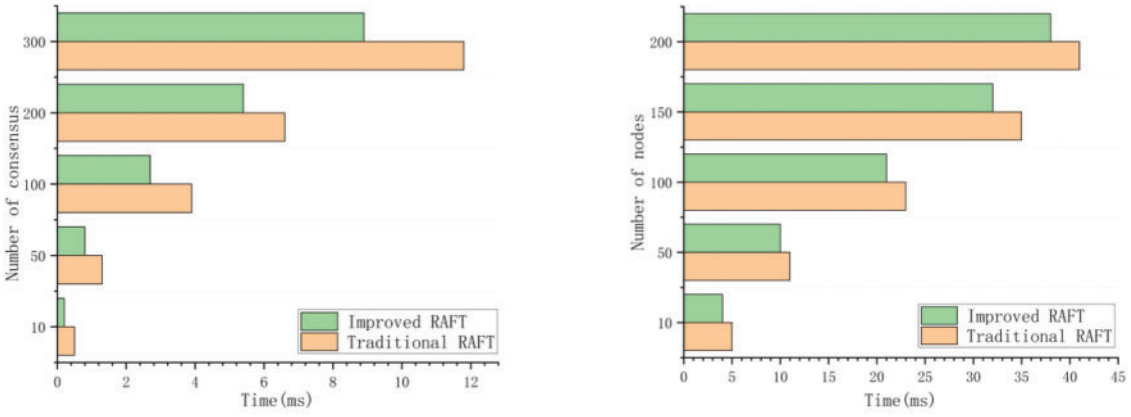
Network environment	File size (Byte)	Write time (MS)	Read time (MS)	Write speed (Mb/s)	Read speed (Mb/s)
LAN	100 M	3365.6	1016.8	29.76	94.33
LAN	10 M	854.2	123	11.76	83.33
LAN	1 M	136.6	34.8	7.35	29.41
Aliyun	100 M	25034	1357	3.94	73.69
Aliyun	10 M	950	145	10.52	68.96
Aliyun	1 M	110	25	9.09	40.00

6.4.3 Evaluation of Selection Efficiency of Improved Raft

Furthermore, in 3.7 Ghz 's I 7-8700CPU and ubuntu18.04-LTS systems, this paper compares the master selection time cost with the traditional Raft algorithm.

The node stability factor in the credit score is calculated according to formula (1), and the penalty factor is punished by multiple punishment, that is, multiple punishment is carried out according to the cumulative amount of wrong private key distribution.

In the experiment, the credit segment is divided into 5 segments, and the penalty factor is [10,9,8.5,8,7]. The simulation tests are carried out under two conditions: the number of alliance nodes is 50, the number of main selections is increased from 10 to 300, and the number of alliance nodes is increased from 20 to 200. In these two cases, the time difference between the traditional Raft algorithm and the improved Raft algorithm is compared. each group carries out 5 repeated tests, taking the average as the test results, as shown in Fig. 9.



(a) Election time under increasing consensus numbers

(b) Election time under increasing node numbers

Figure 9: Comparison of election time

As can be seen from Fig. 9, the time used to select the leader of the traditional Raft and the improved Raft algorithm fluctuates, and the time of the improved algorithm in this paper is less than that of the traditional Raft algorithm. The traditional Raft and the improved Raft algorithm need a round of interaction, and the communication time of voting between nodes is the main factor. During the interaction, the improved algorithm reduces the size of communication nodes through the comparison of credit values, so it has the advantage of selecting the master time, thus improving consensus efficiency.

7 Conclusions and Future Work

For the security of medical data in distributed storage, in this paper, this paper proposes an access control scheme based on CP-ABE. The use of blockchain realizes the traceability of attribute keys distributed by multiple authorized institutions. The improvement of the Raft protocol improves the efficiency of consensus and promotes the accuracy of attribute private key distribution. The ciphertext search method based on IPFS not only overcomes the problem of limited block capacity but also solves the problem of ciphertext search. Most importantly, as far as we know, our scheme puts forward the trinity access policy of user, user attribute, and file attribute for the first time, and implements attribute revocation in a complete sense based on conversion function and proxy re-encryption tool. Finally, in comparison with the other schemes, our scheme has More functions and higher efficiency.

The future work will encompass three key areas. Firstly, we aim to develop a comprehensive system utilizing FISCO BCOS as a foundation. It will involve designing a user-friendly and intuitive UI, as well as implementing smart contracts to enhance the flexibility of rights management. Secondly, we will invest efforts in exploring more efficient access control structures and outsourcing solutions. The objective is to minimize the computational demands on users' mobile devices and wearables, thereby optimizing performance and resource utilization. Lastly, due to the presence of multiple sensors in medical data, we will focus on investigating techniques for multi-sensor data fusion. This research aims to mitigate system storage and communication overhead while ensuring an uninterrupted user experience.

Acknowledgement: The authors thank the Editor, Associate Editor, and reviewers for providing the useful comments to improve the quality of this study.

Funding Statement: This research was funded by the National Natural Science Foundation of China, Grant Number 62162039, and the Shaanxi Provincial Key R&D Program, China with Grant Number 2020GY-041.

Author Contributions: Y. Lu, data curation, conceptualization, software, validation, writing—original draft. WB. Zhang, data curation, resources, methodology, supervision, writing—review and editing. T. Feng and CY. Liu, funding acquisition, project administration, investigation, writing—review, and editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Y. Lu, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. Brent, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *14th Int. Conf. on Practice and Theory in Public Key Cryptography 2011*, Taormina, Italy, pp. 53–70, 2011.

[2] Z. S. Zhang, W. Zhang and Z. G. Qin, "A partially hidden policy CP-ABE scheme against attribute values guessing attacks with online privacy-protective decryption testing in IoT assisted cloud computing," *Future Generation Computer Systems*, vol. 123, no. 10, pp. 181–195, 2021.

[3] J. H. Wei, W. F. Liu and X. X. Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *IEEE Systems*, vol. 2, no. 12, pp. 1731–1742, 2018.

[4] X. H. Yang, W. J. LI and K. Fan, "A revocable attribute-based encryption EHR sharing scheme with multiple authorities in blockchain," *Peer-to-Peer Networking and Applications*, vol. 16, no. 1, pp. 107–125, 2023.

[5] A. Wu, D. Zheng, Y. H. Zhang and M. L. Yang, "Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing," *Sensors*, vol. 7, pp. 2158, 2018.

[6] 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. 10, pp. e0203225, 2018.

[7] Y. T. Yang, F. Z. He, S. H. Han, Y. Q. Liang and Y. Cheng, "A novel attribute-based encryption approach with integrity verification for CAD assembly models," *Engineering*, vol. 6, no. 7, pp. 787–797, 2021.

[8] J. P. Cruz, Y. Kaji and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 2018, no. 6, pp. 12240–12251, 2018.

[9] E. Chen, Y. Zhu, Z. Y. Zhou, S. Y. Lee, W. E. Wong *et al.,* "Policychain: A decentralized authorization service with script-driven policy on blockchain for Internet of Things," *IEEE Internet of Things*, vol. 9, no. 7, pp. 5391–5409, 2022.

[10] Y. Nakamura, Y. Y. Zhang, M. Sasabe and S. Kasahara, "Exploiting smart contracts for capability-based access control in the Internet of Things," *Sensors*, vol. 20, no. 6, pp. 1793, 2020.

[11] A. Ouaddah, A. A. Elkalam and A. A. Ouahman, "FairAccess: A new Blockchain-based access control framework for the Internet of Things: FairAccess: A new access control framework for IoT," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[12] Y. Y. Zhang, S. Kasahara, Y. L. Shen, X. H. Jiang and J. X. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[13] M. Pirretti, P. Traynor, P. McDaniel and B. Waters, "Secure attribute-based systems," in *Proc. of the 13th ACM Conf. on Computer and Communications Security*, Alexandria, Virginia, USA, pp. 99–112, 2006.

[14] K. Sowjanya and M. Dasgupta, "A ciphertext-policy Attribute based encryption scheme for wireless body area networks based on ECC," *Journal of Information Security and Applications*, vol. 54, pp. 102559, 2020.

[15] C. K. Chaudhary, R. Sarma and F. A. Barbhuiya, "RMA-CPABE: A multi-authority CPABE scheme with reduced ciphertext size for IoT devices," *Future Generation Computer Systems*, vol. 138, pp. 226–242, 2023.

[16] Y. Park, M. H. Jeon and S. U. Shin, "Blockchain-based secure and fair IoT data trading system with bilateral authorization," *Computers, Materials & Continua*, vol. 76, no. 2, pp. 1871–1890, 2023.

[17] X. D. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. of 2000 IEEE Symp. on Security and Privacy*, Berkeley, CA, USA, pp. 44–55, 2000.

[18] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin Heidelberg, Springer, pp. 506–522, 2004.

[19] V. R. Atherine and A. S. Nargunam, "Multi authority ciphertext-policy attribute-based encryption for security enhancement in a cloud storage unit," *Sustainable Energy Technologies and Assessments*, vol. 53, pp. 102556, 2022.

[20] J. Su, L. Y. Zhang and Y. Mu, "BA-RMKABSE: Blockchain-aided ranked multi-keyword attribute-based searchable encryption with hiding policy for smart health system," *Future Generation Computer Systems*, vol. 132, pp. 299–309, 2022.

[21] Y. Liang, L. N. Ge, Z. Wang, G. F. Zhang, J. Y. Xu *et al.,* "Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment," *Journal of Cloud Computing-Advances Systems and Applications*, vol. 12, no. 1, pp. 61, 2023.
[22] P. Wang, B. W. Chen, T. Xiang and Z. M. Wang, "Lattice-based public key searchable encryption with fine-grained access control for edge computing," *Future Generation Computer Systems*, vol. 127, pp. 373–383, 2022.
[23] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai and R. Attia, "PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Computer Networks*, vol. 133, pp. 141–156, 2018.
[24] J. Y. Tao and L. Li, "Practical medical files sharing scheme based on blockchain and decentralized attribute-based encryption," *IEEE Access*, vol. 9, pp. 118771–118781, 2021.
[25] H. Cui, R. H. Deng, J. Lai, G. W. Wu and J. Z. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *10th Int. Conf. on Provable Security*, Nanjing, China, vol. 10005, pp. 19–38, 2016.
[26] Z. Q. Zhang, J. B. Zhang, Y. L. Yuan and Z. Li, "An expressive fully policy-hidden ciphertext policy attribute-based encryption scheme with credible verification based on blockchain," *IEEE Internet of Things*, vol. 9, no. 11, pp. 8681–8692, 2022.
[27] J. L. Hao, C. Huang, J. B. Ni, H. Rong, M. Xian *et al.,* "Fine-grained data access control with attribute-hiding policy for cloud-based IoT," *Computer Networks*, vol. 153, pp. 1–10, 2019.
[28] S. Qiu, J. Q. Liu, Y. F. Shi and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *Science China-Information Sciences*, vol. 60, no. 5, pp. 052105, 2017.
[29] W. Zhang, Z. S. Zhang, H. Xiong and Z. G. Qin, "PHAS-HEKR-CP-ABE: Partially policy-hidden CP-ABE with highly efficient key revocation in cloud data sharing system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 613–627, 2022.
[30] J. S. Khan, J. Ahmad, S. S. Ahmed, H. A. Siddiqa, S. F. Abbasi *et al.,* "DNA key based visual chaotic image encryption," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 2, pp. 2549–2561, 2019.
[31] S. F. Abbasi, J. Ahmad, J. S. Khan and M. A. Khan, "Visual meaningful encryption scheme using intertwinning logistic map," in *Proc. of the 2018 Computing Conf.*, London, UK, vol. 2, pp. 764–773, 2019.
[32] J. S. Khan, J. Ahmad, S. F. Abbasi and S. K. Kayhan, "DNA sequence based medical image encryption scheme," in *2018 10th Computer Science and Electronic Engineering (CEEC)*, pp. 24–29, Colchester, UK, 2018.
[33] X. Li, P. Russell, C. Mladin and C. G. Wang, "Blockchain-enabled applications in next-generation wireless systems: Challenges and opportunities," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 86–95, 2021.
[34] D. N. Yang, I. Doh and K. Chae, "Cell based raft algorithm for optimized consensus process on blockchain in smart data market," *IEEE Access*, vol. 10, pp. 85199–85212, 2022.
[35] D. Huang, X. Ma and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Transactions on Systems Man Cybernetics-Systems*, vol. 50, no. 1, pp. 172–181, 2020.
[36] Q. Xu, C. X. Tan, Z. J. Fan, W. Y. Zhu, Y. Xiao *et al.,* "Secure multi-authority data access control scheme in cloud storage system based on attribute-based signcryption," *IEEE Access*, vol. 6, pp. 34051–34074, 2018.
[37] G. C. Hu, L. Y. Zhang, Y. Mu and X. X. Gao, "An expressive "test-decrypt-verify attribute-based encryption scheme with hidden policy for smart medical cloud," *IEEE Systems*, vol. 15, no. 1, pp. 365–376, 2021.
[38] J. Sun, X. M. Yao, S. P. Wang and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.