

Secure Attribute-Based Access Control for Healthcare Data via Blockchain and Proxy Re-encryption

Cen Gu

Suzhou University of Science and Technology

Hongjie Wu

Hongjie.wu@qq.com

Suzhou University of Science and Technology

ChunLiang Wang

The Second Affiliated Hospital of Soochow University

FanFan Kong

The Second Affiliated Hospital of Soochow University

Luping Wang

Suzhou University of Science and Technology

Research Article

Keywords: large language models, Alzheimer's disease, mild cognitive impairment, systematic review, clinical decision support, patient education

Posted Date: May 19th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-8115200/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Secure Attribute-Based Access Control for Healthcare Data via Blockchain and Proxy Re-encryption

Cen Gu¹, Hongjie Wu^{1,2*}, ChunLiang Wang³, FanFan Kong³,
Luping Wang¹

¹*Suzhou University of Science and Technology, 99 Xuefu Road,
SuZhou, 215004, China.

²Jiangsu Province Key Laboratory of Intelligent Building Energy
Efficiency, 99 Xuefu Road, SuZhou, 215004, China.

³The Second Affiliated Hospital of Soochow University, 1055 SanXiang
Road, SuZhou, 215026, China.

*Corresponding author(s). E-mail(s): Hongjie.wu@qq.com;

Abstract

In modern medical systems, massive amounts of data are generated daily, and their secure sharing is crucial for supporting clinical decision-making and improving diagnostic efficiency. However, medical records often contain sensitive personal information, and any leakage during transmission can lead to severe consequences. To deal with these risks, this study implies a blockchain-based access control framework that integrates an IPFS multi-attribute authorization center with enhanced proxy re-encryption technology. Compared with traditional attribute-based access control, the scheme overcomes limitations such as single authorization, inefficiency, and difficulty in accountability. It effectively resists plaintext, collusion, and attribute selection attacks, while significantly reducing the computational overhead of encryption, re-encryption key generation, re-encryption, and decryption. Experimental results and blockchain performance tests demonstrate that the stated approach serves high stability and scalability, making it a practical reaction for secure medical data sharing.

Keywords: Proxy Re-encryption, Blockchain, Privacy Protection, CP-ABE, IPFS

1 Introduction

In the modern healthcare system, there are many types of medical data such as diagnosis and treatment records, experimental prediction results[1, 2] and genomic data[3], etc. For example, diagnosis and treatment records can assist doctors in making medical decisions and become one of the key elements of medical decisions. For example, diagnosis and treatment records can assist doctors to make medical decisions and become one of the key elements of medical decisions, which greatly improves the efficiency of diagnosis and treatment. However, these medical data[4, 5] are often stored in their respective healthcare organizations to form a data silo, so it is urgent to have an efficient sharing solution to break this situation. Considering that medical data often contains some sensitive information, so we must be mindful of the process of sharing data protection, once the consequences of leakage is unimaginable[6] such as the UnitedHealth Group subsidiary Change Healthcare in the United States in 2024 was subject to a ransomware attack, more than one hundred million people's medical privacy data was stolen, including names, medical insurance information, diagnosis and treatment records, etc. In 2018, ransomware infection in China's Tertiary hospitals was detected in 247 Tertiary hospitals across the country, resulting in system paralysis, data locking, and disruption of hospital operations, among others. Therefore, we need to restrict the sharing process, only authorized people can access these medical data so an efficient and secure access control solution is especially important.

1.1 Traditional Access Control Scheme

Access control, a crucial security measure, can successfully stop unauthorized workers from gaining sensitive data access during the sharing of medical information. Through fine-grained permission management, It makes certain that only those with specific IDs can access the pertinent medical data[7, 8], thus reducing the risk of data leakage, safeguarding patient privacy and security, and also enhancing the compliance and controllability of medical data[9]. The primary components of traditional access control schemes include attribute-based, required, list-based, and role-based access control[10, 11].

Role-Based Access Control (RBAC)[12–14] dedicates users to roles, which are then granted permissions, facilitating consistent and manageable access control. However, RBAC may result in permission overaccumulation due to poor role design, lacks flexibility in dynamic or context-sensitive scenarios, and becomes complex to manage as role hierarchies grow. By defining which users or systems have access to each resource and what they are permitted to do, Access Control Lists (ACL)[15–17] regulate permissions at the resource level. While ACLs offer fine-grained control, the proliferation of rules in large-scale systems can make them challenging to maintain and govern. Mandatory Access Control (MAC)[18, 19] enforces system-wide policies based on users' security levels and data classifications. Because users cannot change permissions, MACs are perfect for high-security environments. However, it is complex to implement and maintain, and improper configuration may affect system functionality. Discretionary Access Control (DAC)[20, 21] permits resource owners to ascertain access permissions. While DAC provides flexibility, it poses security risks as users can

potentially grant permissions to unauthorized parties, making it less suitable for environments requiring strict control. Access requests are examined by Attribute-Based Access Control (ABAC)[22, 23] according to user, resource, operation, and environmental attributes. ABAC offers fine-grained, dynamic control suitable for complex application environments, exceeding RBAC’s static model in flexibility.

1.2 Access Control Based on CP-ABE

To enhance overall system security, recent studies have integrated access control with encryption policies. Depending on whether the access policy is incorporated into the key or the ciphertext, attribute-based encryption (ABE) is separated into two categories: ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE)[24–26]. In CP-ABE, the ciphertext is encrypted under a specified access policy, while the user’s decryption key is tied to a set of attributes. Decryption is possible only if the user’s attributes satisfy the access policy. CP-ABE is particularly suitable for secure, fine-grained access control in complex multi-user environments such as medical data sharing[27].

To safeguard the confidentiality of medical data, Zhou et al.[28] proposed a CP-ABE-based telemedicine diagnosis scheme that enables anonymous patient consultations via identifying attributes and restricts doctors’ access to private records through configurable permissions. Akinyele et al.[29] introduced an attribute-based encryption access control scheme for personal health data on mobile devices, allowing data owners fine-grained control over data access and sharing. For wireless medical sensor networks, Zhu et al.[30] designed a CP-ABE-based data exchange mechanism that assigns access rights based on requester attributes and defines specific access policies, incorporating attribute timestamping to support attribute revocation. However, a common limitation of these ABE-based access control systems is the reliance on a single attribute authority to manage attribute permissions and key distribution. This centralized design presents a single point of failure, which compromises system stability and overall security if the authority is compromised.

To address the single-point authorization issue, Chase et al.[31] stated a multi-authority attribute-based system. However, it still relies on a fully trusted central authority that holds the master key and generates the global public key, serving as the core trust anchor. This creates a critical vulnerability: if the central authority is compromised, the security of all encrypted data is undermined, thus failing to achieve true decentralization. Liu et al.[32] further advised a decentralized multi-authority access control scheme, yet the regulator still plays a dominant role. The system remains only partially decentralized and is susceptible to power centralization, which can weaken both its security and efficiency.

1.3 Blockchain-based Access Control

Given the current surge in popularity of blockchain technology, it has been extensively utilized in the healthcare sector for its advantages of decentralization, data immutability and high transparency. In 2016, Azaria et al.[33] presented MedRec, an electronic medical record management system that creates a decentralized record

management system by storing patient medical record information on the blockchain using distributed ledger technology. This approach preserves data integrity and prevents tampering while facilitating the smooth sharing of patients' medical records among distinctive healthcare providers. Ownership, access rights and data integrity of medical records are managed through Smart Contracts. Smart Contracts[34] are able to automatically enforce pre-defined rules, such as granting third party access to specific medical records only if both the patient and the healthcare provider agree. This automated and transparent permission management mechanism greatly improves the effectiveness and security of data management. However, the scheme adopts the Proof of Work (PoW)[35] consensus mechanism, however, this consensus mechanism is computationally intensive, which makes the system as a whole less efficient.

B. Chen et al.[36] and B.B. Gupta et al.[37] proposed integrating blockchain with CP-ABE to enhance security. To address the limitations of centralized healthcare systems, Wang et al.[38] developed a trusted healthcare data management platform leveraging blockchain smart contracts and attribute-based encryption for flexible access control. Jin et al.[39] combined blockchain, smart contracts, and proxy re-encryption for secure medical information sharing. However, it lacks attribute integration, limiting fine-grained access control, and depends heavily on user activity, which may compromise security. Zhen et al.[40] designed a blockchain-based access control scheme with hybrid on-chain/off-chain storage to reduce expense of storage. However, the absence of hashing and signing the IPFS-generated content identifier (CID) on upload enables potential injection of malicious files, lowering system security. Shao[41] highlighted vulnerabilities in CP-ABE-based access control where unencrypted *metadata* transmission can be exploited by attackers. Chen et al.[42] stated a blockchain-enabled encrypted medical information sharing system that stores encrypted data in public clouds and uses blockchain only for index management, constructed via complex Boolean expressions. Zhuang et al.[43] presented a patient-centric health information exchange framework based on blockchain, but inactive facility nodes prevent data requests until reactivation. Noh et al.[44] proposed a user-centric personal medical record management system using proxy-based encryption and blockchain for auditing. However, cooperation between the proxy and requester could potentially expose the data owner's private key. To mitigate this, Liu et al.[45] and Cai et al.[46] advised a federated blockchain-based healthcare data sharing system with enhanced proxy re-encryption for improved data sharing security.

1.4 Contribution

An attribute access control system according to blockchain is stated in this study and proxy re-encryption of electronic medical records, which addresses the issue with An individual point of failure, the *metadata* security issue, and the storage pressure challenge. It employs a mix of off-chain and on-chain storage strategies to significantly lessen the chain's storage pressure. The data encryptor encrypts the *metadata* consisting of the CID, the hash value of the CID, the access policy, the ciphertext, and the hash value of the ciphertext before uploading it to the blockchain and sends the symmetric key to the data visitor who complies with the access policy through the improved proxy re-encryption technique and the visitor gets the *metadata* and then

goes to the IPFS according to the CID to request for this operation protects the security of *metadata* data well.

1.5 Organization

This paper’s general organization is separated into four major sections. Section 2 displays the preliminary knowledge used in the work. Section 3 analyses the overall scheme, mainly from the overall model of the scheme, the description of the scheme and the specific flow of the scheme. Section 4 examines the scheme’s security. Section 5 gives the performance analysis of the proposed scheme. Finally, we wrap off our work in Section 6.

2 Background

Bilinear mappings are briefly reviewed in this section, access control structures, and Decisional Bilinear Diffie-Hellman (DBDH) security assumptions.

2.1 Bilinear Mapping

In elliptic curve cryptography, a bilinear mapping is a mapping e which elements of two elliptic curve groups are mapped to another group[47]. Suppose we have two groups G_1 and G_2 , and a target group G_T . We define a bilinear mapping as shown in Eq. 1.

$$e : G_1 \times G_2 \rightarrow G_T \quad (1)$$

This mapping has several important properties:

Bilinear: For any $p \in G_1$, $q \in G_2$ and scalars a and b , the bilinear mapping is satisfied in Eq. 2.

$$e(p^a, q^b) = e(p, q)^{ab} \quad (2)$$

Non-degradation: The bilinear mapping is not degenerate, there exists at least one $p \in G_1$ and $q \in G_2$, such that $e(p, q) \neq 1$ (unit element).

Computability: For most implementations, computing bilinear mappings is efficient in practice.

2.2 The DBDH Assumption

The DBDH assumption is a widely adopted adjudicative computational complexity assumption in the bilinear group setting[48], and its security is the theoretical basis for the construction of a wide range of cryptographic primitives such as attribute-based encryption, broadcast encryption and proxy re-encryption. The assumption holds that given $g, g^a, g^b, g^c \in G$ where G is a cyclic group of order prime p , the decision that $T = e(g, g)^{abc}$ is computationally indistinguishable from the computationally indistinguishable fact that T is an arbitrary element in the computation of $T \in G_T$. This assumption is the core complexity basis for guaranteeing the safety of IND-CPA.

2.3 Access Control Structure

Define a set of attributes denoted as $\{P_1, \dots, P_n\}$. The set $A \subseteq 2^{\{P_1, \dots, P_n\}}$ is monotonic if and only if for all B, C , if $B \in A$ and $B \subseteq C$, then $C \in A$. A monotonic access control structure A is a set consisting of non-empty subsets of $\{P_1, \dots, P_n\}$, like $A \subseteq 2^{\{P_1, \dots, P_n\}}$, sets in A are referred to as the collection of authorization attributes, and those that are not in it are known as the collection of non-authorization attributes[49, 50].

2.3.1 Access Control Tree

The encryption key is concealed by the tree structure of the CP-ABE access control tree[51, 52], which is used to provide the data's access policy. While the leaf nodes of the tree indicate particular user properties and their values, the inside nodes typically carry out logical processes (such as **and**, **or**, threshold operations). The access control tree's non-leaf nodes are threshold nodes, which implies that their data can only be decrypted in specific situations like threshold operations. The root node determines the general policy, and the visitor can only get the decryption key if their collection of attributes meets the policy in the tree. For example, the threshold for a non-leaf node in Figure. 1 is 2/3, that means that the node has 3 children with a threshold value of 2. Therefore, the visitor needs to satisfy at least 2 children in order to decrypt the secret value of that non-leaf node.

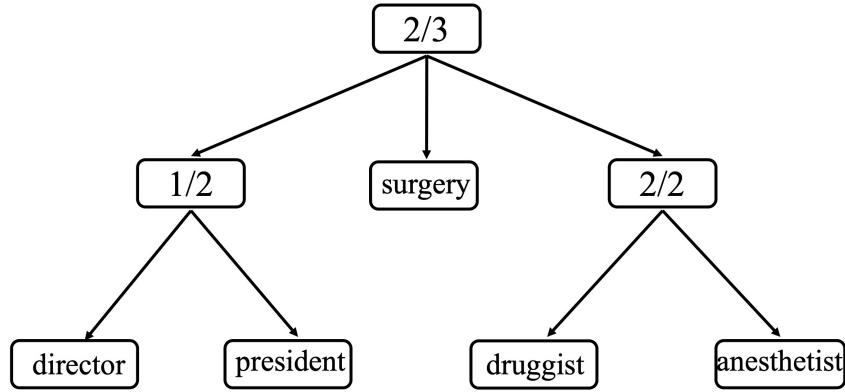


Fig. 1 Access Control Tree

2.3.2 Linear Secret Sharing Scheme

Linear Secret Sharing Scheme (LSSS)[53] first splits the secret into n distinct components and then sends these n parts to n distinct participants, and a set of authorized

participants can recover the secret by merging their shares[54, 55]. It is defined as follows:

The description of the access framework is (M, ρ) , P stands for the group of characteristics, containing the attribute names and attribute values. M is a linear secret sharing matrix of size $l \times n$. ρ is a mapping function, and $\rho(i)$ refers to the mapping of the i th row of the matrix M to an attribute name. Map the secret s to a value on $\mathbb{Z}_p^*$ with p is a prime number. Construct a vector $\mathbf{v} = (s, r_2, r_3, \dots, r_n)^T$ by randomly choosing $\{r_2, r_3, \dots, r_n\} \in \mathbb{Z}_p^*$, and compute $\gamma_i = M_j \cdot \mathbf{v}$ where γ_i refers to the attribute named $\rho(i)$ — the secret share of all uses of the attribute.

Secret reconstruction: Let $B \in M$ be the collection of attributes that comply with the authorizations of the access structure. Where $I = \{i \mid \rho(i) \in B\} \subseteq \{1, 2, \dots, l\}$. Being able to find the coefficients $\{x_i \in \mathbb{Z}_p^*\}_{i \in I}$ in polynomial time such that $\sum_{i \in I} x_i M_i = (1, 0, \dots, 0)$, and then compute $s = \sum_{i \in I} x_i \gamma_i$.

3 Scheme Structure

3.1 System Model

The scheme contains six main components which are medical data owner, medical data user, attribute authorization authority, authority certification authority, star file system and blockchain. The sharing of data is experimented using blockchain technology and smart contracts and the system model diagram of the scheme is displayed in Figure. 2.

Data Owner(DO): Setting the access policy, encrypting the uploaded data, and creating the proxy re-encryption key are the primary responsibilities of the holder of the medical data, which in this system corresponds to the patient.

Data User(DU): The user of the data, which in the context of this scheme means doctors, insurance providers, etc.

Attribute Authorities(AA): Multiple Attribute Authorities work together to manage all attributes, authenticate the identity of the visitor and generate the visitor's attribute key.

Certification Authority(CA): Responsible for the initialization of the system, each attribute authorization authority and user must be registered with a CA and assigned a unique identifier *aid*, *oid* and *uid* for each of them as well as signing the key

Interplanetary File System(IPFS): Mainly responsible for storing encrypted medical data and returning a unique content identifier CID to the DO.

Blockchain(BC): In this scheme is the selected federation chain, which consists of various healthcare institutions and is jointly maintained by each healthcare institution. The federation chain mainly stores the *metadata*, which contains the content unique identifier CID returned by IPFS and the hash value of the DO's hash operation on the CID, which is designed to prevent attackers from modifying the CID and thus causing users to access the malicious files. The *metadata* also contains the access policy formulated by the DO and the hash value of the ciphertext's hash operation, so that visitors can confirm the accuracy of data after getting the ciphertext. In order for the visitor to confirm the safety and integrity of the data after receiving the ciphertext.

The following steps make up the overall scheme's basic process:

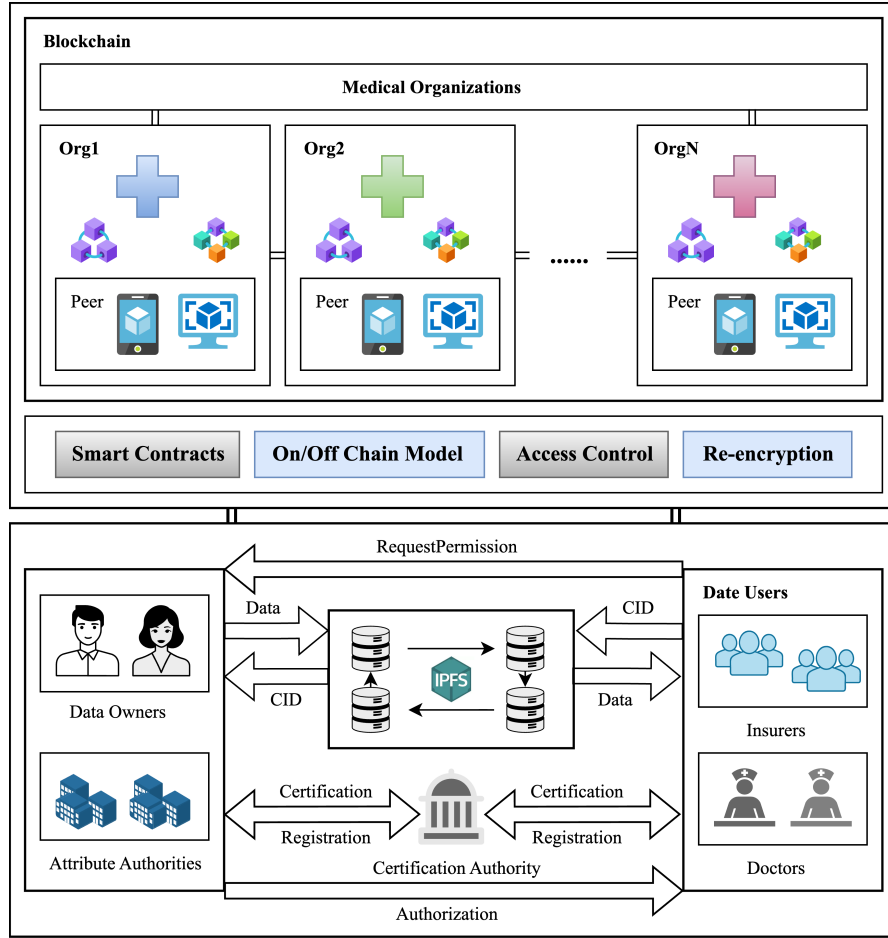


Fig. 2 System Model

Step1: Each entity in the system first registers with CA and obtains its own public and private key, which it uploads to the BC.

Step2: DO submits the data to the IPFS system after encrypting it, which returns the unique identifier CID of the ciphertext to DO, considering the security DO hashes the CID and builds the *metadata* for the CID, the CID hash, the ciphertext CT, the ciphertext's hash, and the access policy to be uploaded to the blockchain.

Step3: DU requests an access request from DO, and DO verifies that the DU's collection of attributes conforms to the access policy.

Step4: If the access policy is met, AA builds an attribute private key for the DU, and the DO builds an agent re-encryption key for the DU and uploads it to the BC.

Step5: BC re-encrypts the *metadata* using the proxy re-encryption key and sends it to the DU. Using its own private key, the DU decrypts the data, hashes the CID, and checks the hash value with the CID in the *metadata*; if they match, the data has not been altered.

Step6: Based on the CID, DU asks IPFS for access to the CT. It hashes the CT to guarantee data integrity before decrypting it using the attribute private key for data access.

3.2 Scheme Definition

3.2.1 System Initialization Phase

Step1: CA accepts security parameter λ as input parameter and produces global public parameters GPP , signature key CA_{sk} and verification key CA_{pk} .

Step2: CA registers each AA with the algorithm inputting the global public parameters GPP and the attribute authorization authority's identity information $info_a$, and outputting the AA's unique identity aid and authentication key CA_{pk} , as well as the AA's master public key PK_{aid} and master private key SK_{aid} .

Step3: CA registers the identity of the data owner, the algorithm inputs global public parameter GPP , the signing key CA_{sk} and owner's identity information $info_o$. It results the unique identity of the data owner oid , public key PK_{oid} and private key SK_{oid} , and the certification of authorization $Cert_{oid}$ for DO.

Step4: CA registers the identity of DU, the algorithm inputs the global public parameter GPP , the signing key CA_{sk} and user's identity information $info_u$. The algorithm outputs the data owner's unique identity uid , public key PK_{uid} and private key SK_{uid} , and the certification of authorization $Cert_{uid}$ for DU.

3.2.2 Key Generation Phase

AA is in charge of carrying out the algorithm, and the data user u_{uid} submits to the attribute authorization authority the collection of qualities that are part of the authority management. The algorithm takes a user authorization certificate $Cert_{uid}$ as input, a global public parameter GPP , a private key of the attribute authority SK_{aid} , a collection of attributes of the user belonging to the attribute authority $S_{uid,aid}$, and an authentication key of the CA_{pk} , and outputs an attribute private key $SK_{uid,att}$ of the data user u_{uid} .

3.2.3 Data Encryption Phase

Considering the efficiency of encrypting a large amount of data, the original record is encrypted using a symmetric encryption algorithm using a symmetric key k that is chosen at random by DO. The secret key k is then encrypted using an access policy (M, ρ) to produce the CT embedded with the access policy.

3.2.4 Data Decryption Phase

An authorized user receives *metadata* from the healthcare federation chain and obtains the ciphertext CT from IPFS based on the CID in the *metadata* and decrypts it with his or her own attribute private key to acquire the symmetric key k , and then decrypts the healthcare data with the symmetric key k to pick up the plaintext.

3.3 Definition of Security

Indistinguishability under the flexible Choice Ciphertext Attack (IND-CCA2) is the foundation of this scheme's safety, which is mainly satisfied in the current design by Choice Plaintext Attack (IND-CPA). We adopt the standard security game framework outlined as follows:

Initialization: The challenger generates the system parameters, the adversary $\mathcal{A}$ decides the collection of corrupted attribute authorization agencies $\mathcal{C} \subseteq \{AA_1, \dots, AA_N\}$, the challenger induces the public key of the uncorrupted agencies $\{PK_{aid}\}_{aid \notin \mathcal{C}}$ and leaks the master key of the corrupted agencies $\{MSK_{aid}\}_{aid \in \mathcal{C}}$ to the adversary $\mathcal{A}$.

Phase 1: Adversary $\mathcal{A}$ can adaptively query the private key of an arbitrary collection of attributes S . S does not meet the challenge strategy $\mathcal{A}^*$.

Challenge Phase: Adversary $\mathcal{A}$ offers two equal-length plaintexts m_0, m_1 and a challenge strategy $\mathcal{A}^*$. The challenger selects $\mathcal{R} \in \{0, 1\}$ at random, produces a ciphertext $CT^* = \text{Encrypt}(m_{\mathcal{R}}, \mathcal{A}^*)$ and returns it to $\mathcal{A}$.

Phase 2: Adversary $\mathcal{A}$ can keep on querying the private key with the same constraints as in Phase 1.

Guessing Phase: Adversary $\mathcal{A}$ outcomes a guess $\mathcal{R}' \in \{0, 1\}$, and triumphs if $\mathcal{R}' = \mathcal{R}$. The scheme is safe under IND-CPA if, for any polynomial time adversary $\mathcal{A}$, its dominance meets Eq. 3.

$$Adv_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) = \left| \Pr[\mathcal{R}' = \mathcal{R}] - \frac{1}{2} \right| \leq \text{negl}(\lambda) \quad (3)$$

3.4 Construction

The key generation, encryption, decryption, and system startup phases comprise the entirety of the system model. The particular flow of each phase is explained as follows:

3.4.1 Initialization

a. CA.Setup: CA executes the *CA.Setup* algorithm to initialize the system. Inputting the security parameter λ and choosing two orders as prime p , the multiplicative cyclic groups G and G_T with elements g are generated as well as the bilinear mapping $e : G \times G \rightarrow G_T$. Choosing three hash functions $H : \{0, 1\}^* \rightarrow G$. The final output of the global common parameter *GPP* is shown in Eq. 4.

$$GPP = \{G, G_T, g, p, H, e\} \quad (4)$$

b. AA.Register: Each AA registers with CA, AA sends registration request to CA and sends AA's identity information $info_a$. CA generates unique identity $aid \leftarrow H(info_a)$ for AA, then AA randomly selects $\alpha_{aid}, \beta_{aid} \in \mathbb{Z}_p$ as its own private key SK_{aid} , generates corresponding public key $PK_{aid} = (g^{\beta_{aid}}, e(g, g)^{\alpha_{aid}})$ and register its public key with CA.

c. DO.Register: Each DO registers with CA, DO sends a registration request and sends the CA his own identity information $info_o$, CA generates a unique identity $oid \leftarrow H(info_o)$ for the DO, the DO randomly selects $\gamma \in \mathbb{Z}_p$ as his own private key

SK_{oid} , generates the corresponding public key $PK_{oid} = g^\gamma$ and register his public key with CA.

d. DU.Register: Each DU registers with CA, DU initiates the registration request and sends its identity information $info_u$ to CA, which builds a unique identity $uid \leftarrow H(info_u)$ for the DU, and the DU randomly selects $\mu \in \mathbb{Z}_p$ as its private key SK_{uid} , generates the corresponding public key $PK_{uid} = g^\mu$ and register their public key with CA.

3.4.2 Key Generation

KeyGen: The user requests a private key from each attribute authorization authority AA_{aid} that manages his attributes, assuming that the collection of attributes of the user is S and the attribute $att \in S$ is managed by the authority AA_{aid} . Before that, the attribute authority will verify the authenticity of the user's identity uid and the legitimacy of the $Cert_{uid}$. If the verification passes then the attribute authorization authority will randomly select $r_{aid} \in \mathbb{Z}_p$ to build the attribute private key component for the user as shown in Eq. 5.

$$SK_{att} = (g^{\alpha_{aid}} \cdot H(att)^{r_{aid}}, g^{r_{aid}}) \quad (5)$$

Then the final private key of the user is the set of private keys produced by all relevant attribute authorizations as shown in Eq. 6.

$$SK_{uid,att} = \bigcup_{aid} \{SK_{att} \mid att \in S \cap AA_{aid}\} \quad (6)$$

3.4.3 Encryption

Enc: To ensure the efficiency of encrypting a large amount of data, the DO randomly picks a symmetric key $k \in \mathbb{G}_T$ to encrypt a substantial amount of medical data to acquire the ciphertext C . The LSSS is used to formulate the access policy (M, ρ) to encrypt the symmetric key k . Where M is a matrix of $l \times n$ and ρ is a mapping function to map the rows of the matrix to an attribute. Constructing random vector $\mathbf{v} = (k, v_2, \dots, v_n) \in \mathbb{Z}_p^*$ firstly. Computing the secret share as in Eq. 7, each x_i is bound to an attribute $\rho(i)$, and for each row i , assuming that $\rho(i)$ is managed by the organization AA_{aid} , then a random $t_i \in \mathbb{Z}_p$ computes the ciphertext component as in Eq. 8, and calculate the pairing value as Eq. 9, the final combination of CID and CID's hash value, along with the ciphertext's hash value, forms *metadata*. is shown in Eq. 10.

$$x_i = M_i \cdot \mathbf{v}, \forall i \in \{1, \dots, l\} \quad (7)$$

$$C_{i,1} = g^{\beta_{aid} x_i}, C_{i,2} = H(\rho(i))^{x_i}, C_{i,3} = g^{t_i} \quad (8)$$

$$D_i = e(g, g)^{\alpha_{aid} x_i} \cdot e(g, g)^{\beta_{aid} t_i} \quad (9)$$

$$metadata = \left(C, (M, \rho), \{C_{i,1}, C_{i,2}, C_{i,3}, D_i\}_{i=1}^l, k \cdot \prod_{aid} e(g, g)^{\alpha_{aid}k} \right) \quad (10)$$

Then DO encrypts the *metadata* with the randomly selected symmetric key $s \in G_T$, and then uploaded to the BC, and with the DO randomly picks $h \in \mathbb{Z}_p^*$ and employs proxy re-encryption technique to encrypt the symmetric key s according to Eq. 11 to Eq. 13 and uploads it to the healthcare federation chain.

$$CT_1 = g^h \quad (11)$$

$$CT_2 = s \cdot e(PK_{oid}, g)^h = s \cdot e(g, g)^{\gamma h} \quad (12)$$

$$CT_s = (CT_1, CT_2) \quad (13)$$

3.4.4 Decryption

Dec: When DU meets the access policy, the DO produces a proxy re-encryption key $RK_{O,U}$ for the DU based on its own private key and the public key of DU according to Eq. 14 and conveys it to BC. BC accepts the proxy re-encryption key, it creates a proxy re-encryption ciphertext CT_R for the DU that complies to the access policy according to Eq. 15 and sends it to the corresponding DU.

$$RK_{O,U} = PK_{uid}^{\frac{1}{SK_{oid}}} = g^{\frac{\mu}{\gamma}} \quad (14)$$

$$CT_R = (CT'_1, CT'_2) = (e(CT_1, RK_{O,U}), CT_2) \quad (15)$$

DU receives the proxy re-encrypted ciphertext and uses its private key to decrypt the encrypted key according to Eq. 16.

$$s = \frac{CT'_2}{CT'_1^{\frac{1}{\mu}}} = \frac{s \cdot e(g, g)^{\gamma h}}{e(CT_1, RK_{O,U})^{\frac{1}{\mu}}} = \frac{s \cdot e(g, g)^{\gamma h}}{e(g^h, g^{\mu\gamma})^{\frac{1}{\mu}}} = \frac{s \cdot e(g, g)^{\gamma h}}{e(g, g)^{\gamma h}} \quad (16)$$

DU decrypts the *metadata* after getting s . After decryption, it first hashes the CID in the *metadata* and verifies the uniqueness. If the verification is consistent, DU submits a request for access to IPFS based on the CID, and IPFS sends CT to DU after receiving the request. After getting the CT, DU first hashes the CT and verifies the uniqueness. If DU meets the access policy, in addition to a reconstruction factor $\{w_i\}$ such that $\sum_{i \in I} w_i M_i = (1, 0, \dots, 0)$. The pairwise value of each share is recovered through Eq. 17 and use the reconstruction coefficients w_i to decrypt the symmetric key k according to Eq. 18 and Eq. 19. Eventually the DU decrypts the C via the obtained symmetric key k to get the data it wants to access.

$$\frac{e(C_{i,1}, g^{\alpha_{aid}} H(\rho(i))^{r_{aid}})}{e(C_{i,3}^{\beta_{aid}}, g^{r_{aid}})} = \frac{e(g^{x_i}, g^{\alpha_{aid}} H(\rho(i))^{r_{aid}})}{e(g^{\beta_{aid} t_i}, g^{r_{aid}})} \quad (17)$$

$$\prod_{i \in I} (e(g, g)^{\alpha_{aid} \lambda_i})^{w_i} = e(g, g)^{\sum_{i \in I} \alpha_{aid} \lambda_i w_i} = e(g, g)^{\sum_{aid} \alpha_{aid} k} \quad (18)$$

$$k = \frac{k \cdot \prod_{aid} e(g, g)^{\alpha_{aid} k}}{e(g, g)^{\sum_{aid} \alpha_{aid} k}} \quad (19)$$

4 Security Analysis

4.1 Correctness Analysis

We employ an attribute-based encryption technique in this study, so the correctness of the scheme requires that the key can be recovered and the ciphertext can be accurately decrypted and the plaintext recovered when and only when the user's collection of attributes meets the access policy defined by the ciphertext. The access policy is (M, ρ) , where M is an $l \times n$ matrix and ρ is a mapping function serves to map the rows of the matrix to specific attributes. In this study, we adopt a multi-attribute authorization authority scheme, where different attributes of data visitors belong to different attribute authorization authorities for management.

For each attribute $\rho(i)$ and management authority AA_{aid} to which it belongs, the private key of the attribute is created as

$$SK_{\rho(i)} = (g^{\alpha_{aid} H(\rho(i))^{r_{aid}}}, g^{r_{aid}}) \quad \text{and} \quad H(\rho(i)) = g^{\xi_{\rho(i)}},$$

which is randomly chosen at the time of encryption as $t_i = \xi_{\rho(i)} x_i$, and computed as the corresponding cipher text component $C_{i,1} = g^{x_i}, C_{i,3} = g^{t_i}$. Then the derivation proof process of $e(g, g)^{\alpha_{aid} x_i}$ in the decryption process is shown in Eq. 20.

$$\begin{aligned} \frac{e(C_{i,1}, SK_{\rho(i),1})}{e(C_{i,3}, SK_{\rho(i),2})} &= \frac{e(g^{x_i}, g^{\alpha_{aid} H(\rho(i))^{r_{aid}}})}{e(g^{t_i}, g^{r_{aid}})} \\ &= \frac{e(g^{x_i}, g^{\alpha_{aid}}) \cdot e(g^{x_i}, H(\rho(i))^{r_{aid}})}{e(g^{t_i}, g^{r_{aid}})} \\ &= \frac{e(g, g)^{x_i \alpha_{aid}} \cdot e(g, g)^{x_i \xi_{\rho(i)} r_{aid}}}{e(g, g)^{x_i \xi_{\rho(i)} r_{aid}}} \\ &= e(g, g)^{\alpha_{aid} x_i} \end{aligned} \quad (20)$$

When the attributes of DU reach the access policy, then the k is recovered as in Eq. 21 and 22, where w_i is the reconstruction factor. This shows that DU can recover the symmetric key correctly.

$$\prod_{i \in I} (e(g, g)^{\alpha_{aid} x_i})^{w_i} = e(g, g)^{\sum_{i \in I} \alpha_{aid} x_i w_i} = e(g, g)^{\alpha_{aid} k} \quad (21)$$

$$k = \frac{k \cdot e(g, g)^{\sum_{aid} \alpha_{aid} k}}{e(g, g)^{\sum_{aid} \alpha_{aid} k}} \quad (22)$$

4.2 Proof of Security

Theorem: There is an algorithm $\mathcal{B}$ that can handle the DBDH problem with the same benefit if there exists an adversary $\mathcal{A}$ that challenges the IND-CPA safety of this scheme with a non-negligible profit ϵ .

Proof: Given a challenge tuple (g, g^a, g^b, g^c, T) , where $T = e(g, g)^{abc}$ or a arbitrary quantity $r \in G_T$, the ultimate objective is to determine whether T is $e(g, g)^{abc}$ or not. Construct a simulator $\mathcal{B}$ and send the challenge tuple (g, g^a, g^b, g^c, T) to the simulator $\mathcal{B}$, and use the production of the adversary $\mathcal{A}$ to judge whether T is $e(g, g)^{abc}$.

System initialization: Simulator $\mathcal{B}$ sets the public parameter g and hash function H modeled as an arbitrary prophecy machine. For each agency to build their own public key, if the attribute authorization agency $AA_{aid} \notin \mathcal{C}$, then randomly select $\alpha_{aid} \in \mathbb{Z}_p$, implicitly set $\beta_{aid} = a$ embedded in the DBDH parameter, then the public key is shown in Eq. 23.

$$PK_{aid} = (g^{\beta_{aid}} = g^a, e(g, g)^{\alpha_{aid}}) \quad (23)$$

If $AA_{aid} \in \mathcal{C}$, then randomly choose $\alpha_{aid}, \beta_{aid} \in \mathbb{Z}_p$, the public key is shown in Eq. 24.

$$PK_{aid} = (g^{\beta_{aid}}, e(g, g)^{\alpha_{aid}}) \quad (24)$$

and leak the master key $MSK_{aid} = (\alpha_{aid}, \beta_{aid})$ to the adversary $\mathcal{A}$.

Hash prediction machine simulation: Hash query $H(att)$ for the adversary, if att has not been queried, then randomly choose $\delta_{att} \in \mathbb{Z}_p$, record (att, δ_{att}) and return $H(att) = g^{\delta_{att}}$; otherwise directly return the recorded $H(att)$.

Private key query: When the adversary $\mathcal{A}$ inquiries the private key of the attribute collection S , if S meets the challenge policy $\mathcal{A}^*$ then terminates the game, the adversary is invalidated. Otherwise, for each attribute $att \in S$ managed by the attribute authorization authority AA_{aid} : If $AA_{aid} \notin \mathcal{C}$, then randomly select $r_{aid} \in \mathbb{Z}_p$ and compute the private key component as shown in Eq. 25. Otherwise, if $AA_{aid} \in \mathcal{C}$, then directly use corrupted MSK_{aid} to generate private key.

$$SK_{att} = (g^{\alpha_{aid}} \cdot H(att)^{r_{aid}}, g^{r_{aid}}) \quad (25)$$

Challenge ciphertext generation: Set the symmetric key $k = m_{\mathcal{R}} \cdot T$ where T is the DBDH challenge value, and then construct a random vector $\mathbf{v} = (k, v_2, \dots, v_n)$ to compute the secret share $x_i = M_i \cdot \mathbf{v}$. The attribute att_i corresponding to each row i is managed by the attribute authorizer AA_{aid} . If $AA_{aid} \notin \mathcal{C}$, then set $t_i = \delta_{att_i} \cdot x_i$, and calculate the ciphertext component and pairing value as shown in Eq. 26. Then the final ciphertext can be expressed as shown in Eq. 27.

$$\begin{aligned} c_{i,1} &= (g^a)^{x_i}, & c_{i,3} &= g^{t_i} = g^{\delta_{att_i} \cdot x_i}, \\ D_i &= e(g, g)^{\alpha_{aid} x_i} \cdot e(g^a, g^{t_i}) = e(g, g)^{\alpha_{aid} x_i + at_i} \end{aligned} \quad (26)$$

$$CT^* = \left(M, \rho, \{c_{i,1}, c_{i,2} = H(att_i)^{x_i}, c_{i,3}, D_i\}, k \cdot \prod_{aid \notin \mathcal{C}} e(g, g)^{\alpha_{aid} k} \right) \quad (27)$$

Adversary guess and statute: If adversary $\mathcal{A}$ outputs $\mathcal{R}' = \mathcal{R}$, then $\mathcal{B}$ determines that $T = e(g, g)^{abc}$, otherwise T is An arbitrary. When $T = e(g, g)^{abc}$ then $k \cdot T = m_{\mathcal{R}} \cdot e(g, g)^{abc} = m_{\mathcal{R}} \cdot e(g^a, g^{bc})$, at this time, the challenge ciphertext is legal encryption, the adversary $\mathcal{A}$ advantage ϵ direct transmission. When T is a random value of $\mathcal{F}$ then $k \cdot \mathcal{F} = m_{\mathcal{R}} \cdot \mathcal{F}$, adversary $\mathcal{A}$ cannot obtain information about k and the advantage is 0. Therefore, $\mathcal{B}$ works out the DBDH obstacle with the same benefit ϵ as the adversary.

4.3 Specific Attack Scenarios and Defense Mechanisms

4.3.1 Collusion Attack Defense

Suppose the private keys of user A and user B are shown in Eq. 28. The private key after collusion generation is shown in Eq. 29. The cross-computation at the time of decryption is shown in Eq. 30. Since $t_i = \delta_{att} \cdot x_i$ and $H(att) = g^{\delta_{att}}$, the numerator and denominator are equal crosswise to 1, but the master key after aggregation is $2\alpha_{aid}$, which does not match with the original master key α_{aid} , resulting in the failure to recover $e(g, g)^{\alpha_{aid}K}$ correctly.

$$SK_{att}^{(A)} = (g^{\alpha_{aid} H(att) r_{aid}^{(A)}}, g^{r_{aid}^{(A)}}), SK_{att}^{(B)} = (g^{\alpha_{aid} H(att) r_{aid}^{(B)}}, g^{r_{aid}^{(B)}}) \quad (28)$$

$$SK_{att}^{(A+B)} = (g^{2\alpha_{aid} H(att) r_{aid}^{(A)} + r_{aid}^{(B)}}, g^{r_{aid}^{(A)} + r_{aid}^{(B)}}) \quad (29)$$

$$\frac{e(g, H(att))^{\beta_{aid} x_i (r_{aid}^{(A)} + r_{aid}^{(B)})}}{e(g, g)^{\beta_{aid} t_i (r_{aid}^{(A)} + r_{aid}^{(B)})}} \quad (30)$$

4.3.2 Attribute Selection Attack

If adversary $\mathcal{A}$ wants to pre-select a specific attribute to construct a conflict in an attempt to bypass the access policy. Then the row random numbers $v_2, \dots, v_n$ of the LSSS matrix ensure that adversary $\mathcal{A}$ cannot predict x_i by selecting an attribute, and the collision-resistant hash function makes it impossible for adversary $\mathcal{A}$ to find $att \neq att'$ that satisfies $H(att) = H(att')$, and thus cannot forge an attribute authorization.

5 Performance Analysis

Every experiment in this work was carried out on a device equipped with an Apple M4 processor, 16GB RAM, and macOS Sequoia 15.5 operating system. The specific environment configuration is Hyperledger Fabric v2.5.12, Docker v28.3.2, Docker-Compose v2.39.2, Node v18.20.8 and IPFS v0.35.0.

5.1 Functional Analysis

As demonstrated in Table 1, the proposed scheme in this paper has some assists over existing schemes in relation to patient-centricity, identity information protection, data confidentiality, secure distributed storage and fine-grained access. Ref.[36] and Ref.[46] ignore identity privacy protection in their schemes, Ref.[37] relies on the dependence of healthcare data on cloud storage, which has potential data security risks, and Ref.[38] employs trusted organization-centric access control, which restricts the patient’s control over the ownership of healthcare data. Ref.[39] uses an identity-based access control scheme that does not achieve fine-grained access well. Therefore, the scheme in this paper has some advantages in these aspects.

Table 1 Comparison of Existing Schemes

Aspect	Scheme					Ours
	[36]	[37]	[38]	[39]	[46]	
Patient-centered	✓	✓	✓	✗	✗	✓
Data Confidentiality	✓	✓	✓	✓	✓	✓
Identity Privacy Protection	✗	✗	✓	✗	✓	✓
Secure Distributed Storage	✗	✗	✓	✓	✓	✓
Fine-grained Access Control	✓	✗	✓	✗	✗	✓

The “✓” and “✗” symbols represent supported and unsupported features, respectively.

5.2 Calculation Cost Analysis

In this study, we use Hyperledger Caliper (v0.6.0) as the benchmarking tool for performance testing. Caliper is specifically designed to examine the blockchain systems’ performance.

In this experiment, we conducted latency tests on four distinct smart contracts across four distinct phases: registration, data upload, authorization request, and data access. As shown in Table 2, During load testing of the three smart contract categories, the system maintained stable efficiency: except for slight fluctuations in isolated use cases under extremely high concurrency such as uploadData and requestPermission with *Valid* $\approx$ 98–99.7%, all other scenarios achieved 100% efficiency. The latency of registerUser was nearly insensitive to total volume and injection rate: *Avg* $\approx$ 2030–2032 ms, *p50* $\approx$ 2030–2032 ms, *p95* $\approx$ 2038–2041 ms, *p99* $\approx$ 2040–2048 ms, with 100% success under all loads. This compact quantile distribution indicates that bottlenecks are primarily determined by block confirmation or consensus rounds, rather than local computation or storage. Consequently, this control plane operation exhibits block-time-dominated, predictable, and stable latency characteristics. However, user-perceived registration delays remain around 2 seconds, suggesting that asynchronous confirmation or shorter block times or faster confirmation strategies could be adopted to optimize the experience.

In contrast, `uploadData` and `requestPermission` belong to data-plane operations and exhibit more typical load-dependent latency. As the request rate increased from 10 to 150, the average latency gradually rose from $\approx 50\text{--}70$ ms to $\approx 130\text{--}180$ ms, while $p95/p99$ grew from $\approx 80\text{--}120$ ms to $\approx 220\text{--}300$ ms. Even under the high-stress scenario with $total = 3000$ and $rate = 150$, both operations maintained $p99 \leq 250\text{--}270$ ms and efficiency $\geq 99\%$. This “moderate mean increase with controlled but amplified tail” pattern indicates that resource contention exists on the data plane such as external storage or network I/O. Nevertheless, the system demonstrates scalability with graceful degradation, without collapse-like growth or large-scale failures. Overall, these results demonstrate that the stated cloud data sharing scheme achieves strong determinism and stability in the control plane, while providing low tail latency and high success rates in the data plane under high throughput. This meets the key requirements of practical deployment for business predictability and scalability. To visually illustrate the changes, this paper selected the most representative indicator P95 to plot three line charts as shown in Figure. 3, (a) represents the chaincode `registerUser`, (b) represents the chaincode `uploadData`, and (c) represents the chaincode `requestPermission`. Additionally, we conducted throughput tests on the three types of chain codes at different transmission rates and compared them with Ref.[39], with the results shown by (b) in Figure. 3. It is evident that as the transmission rate increases from 50 TPS to 200 TPS, the throughput of all three chain codes showed a clear upward trend. For instance, the Register chain code increased from 42.50 TPS to 161.17 TPS. This contrasts with the results in Ref.[39], where the throughput of the chain code remained relatively stable across different transmission rates and did not demonstrate improvement with increasing concurrent requests. Our experimental results demonstrate that the designed chaincode exhibits excellent scalability and resource utilization under the current network configuration, capable of continuously increasing throughput under high concurrent requests, highlighting the system’s performance advantages.

In the `viewData` chaincode, we conducted load testing on each of the four methods within the chaincode. We conducted load testing on each of its four methods. The load testing results for the `viewDate` and `Confirm` methods are shown in Table 3, while the results for `ViewAndLog` and `ViewBundle` are presented in Table 4. We also selected the most representative p95 metrics to plot a line chart, as shown in Figure. 4. The categories (a), (b), (c), and (d) correspond to the four method types: `requestView`, `Confirm`, `ViewAndLog`, and `ViewBundle`. Experimental results demonstrate that the four methods of the `viewData` chaincode exhibit excellent stability under varying transaction volumes and request rates. Latency increases smoothly and linearly with load, showing no abnormal jitter, reflecting strong robustness. Regarding scalability, latency performance was comparable across methods under low load. While latency differences amplified for some methods under high load, overall performance remained within manageable limits, validating the solution’s robust scalability. Additionally, elevated average latency in specific scenarios primarily stemmed from inherent blockchain characteristics: queueing effects, resource contention, and long-tail delays—unavoidable system traits in distributed environments.

Additionally, this paper employs an improved proxy re-encryption technique that reduces computational overhead by minimizing exponential operations and eliminating

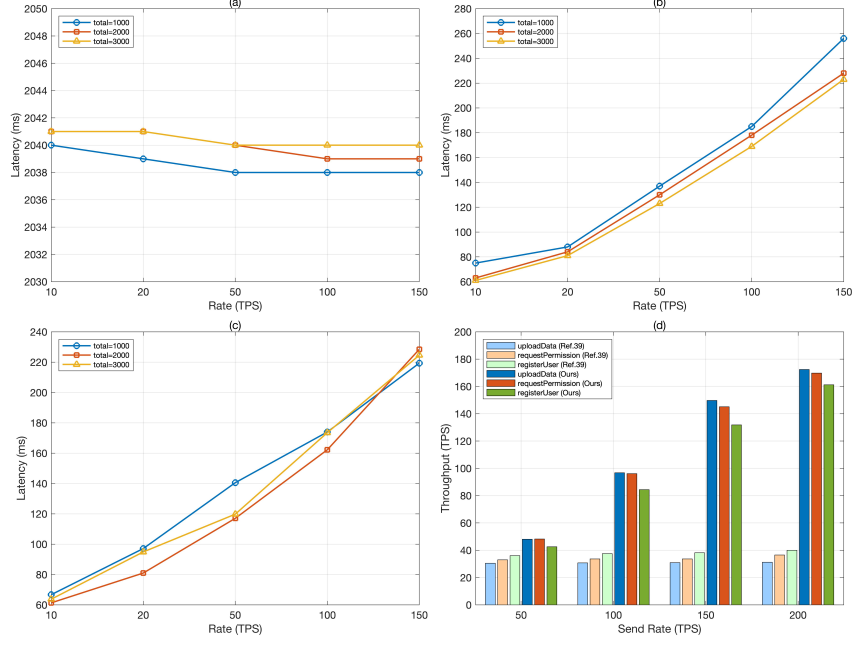


Fig. 3 P95 Metrics for Under Different Total and Rate Conditions and Throughput Comparison

hash computations. It is also compared with the schemes in Ref.[45] and Ref.[46], Table 5 displays the comparison findings, assuming that E stands for the exponential operation in group $\mathbb{G}$, L for the bilinear pairing operation, and H for the hash operation. The medical data selected in this paper are of sizes 100MB, 200MB, 400MB, 600MB, and 800MB. These datasets were encrypted using a symmetric encryption algorithm, with the corresponding encryption times shown by (a) in Figure. 5. Subsequently, we employed an improved proxy re-encryption technique to re-encrypt the symmetric encryption key s used for encrypting *metadata*. Comparisons of the time overhead across each stage of re-encryption with existing re-encryption schemes demonstrated certain advantages, as illustrated in (b) of Figure. 5. Furthermore, the time overhead of uploading the encrypted secret message to the IPFS network and obtaining the secret message from the IPFS network by DU according to the CID is demonstrated by in Figure. 6. The heatmap reveals that this solution generally exhibits darker colors for both IPFS upload and download durations compared to Ref.[39], indicating a significant advantage in time efficiency for this aspect and the CPU utilization of the program execution during uploading and obtaining is demonstrated by in Figure. 7. After uploading data to IPFS and obtaining the CID, DO must perform a hash operation on the CID and ciphertext. This information is then combined into *metadata* and uploaded to the blockchain. Therefore, we recorded the time consumed by the hashing operation as shown in Table 6. The labels 1, 2, 3, 4, and 5 correspond to five different record sizes and their respective CID. The data in the table indicates that even when

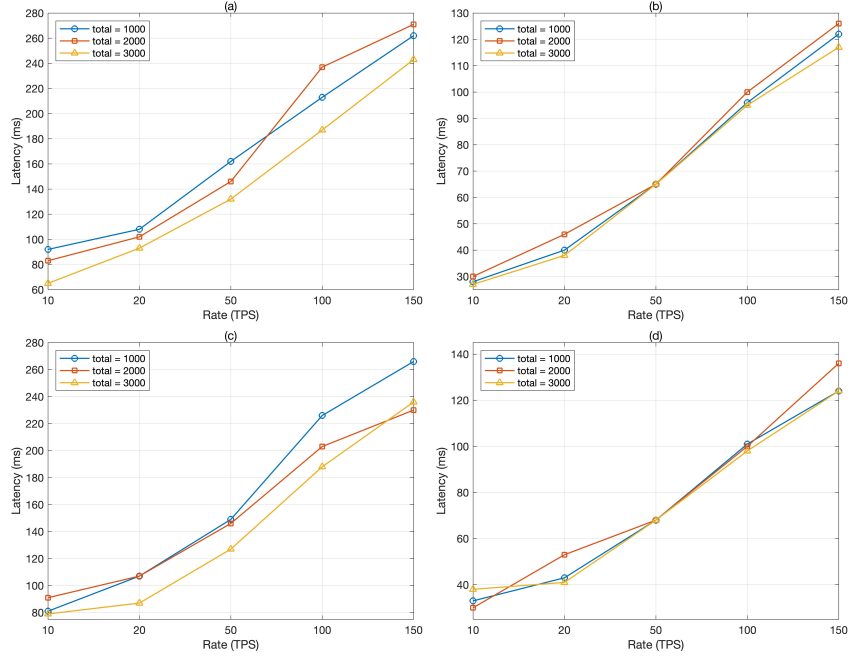


Fig. 4 P95 metrics for viewData under different total and rate conditions

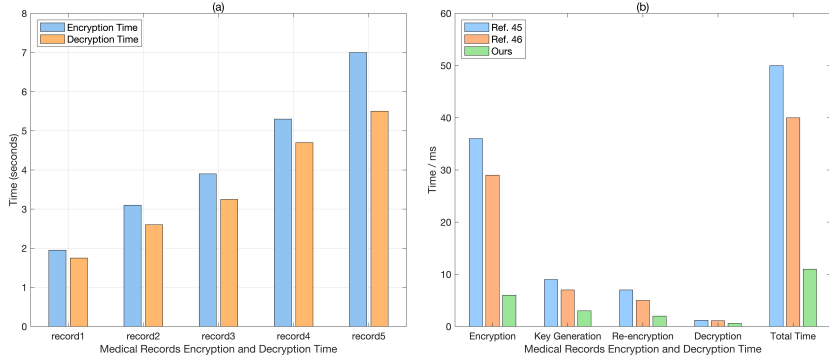


Fig. 5 Medical records encryption and decryption time, and proxy re-encryption time overhead comparison

the record size reaches 800MB, the operation can still be completed within a few seconds. The time spent hashing the CID consistently remains at the millisecond level, demonstrating high efficiency.

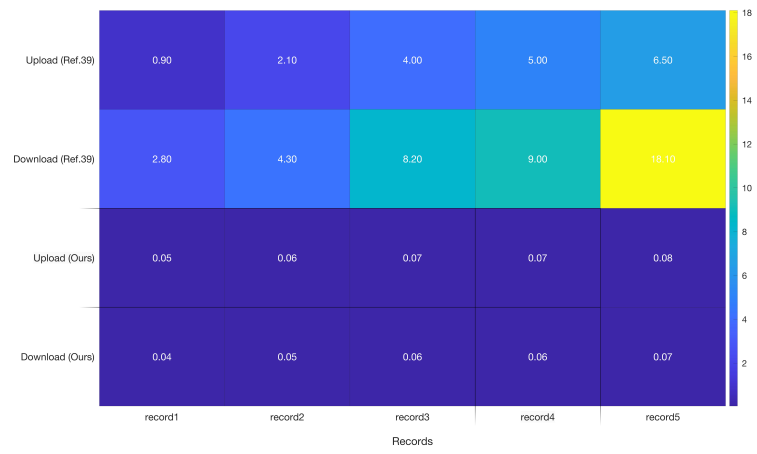


Fig. 6 Upload and download time comparison

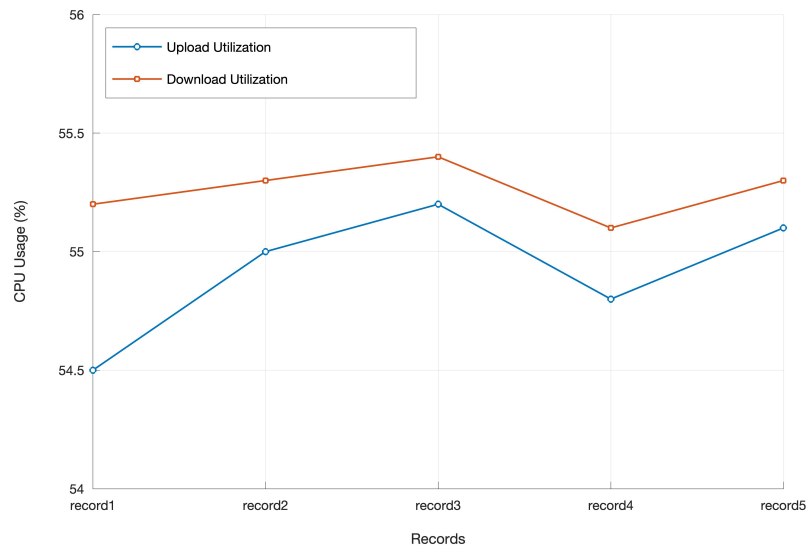


Fig. 7 CPU utilization comparison

Table 2 End-to-end performance under different totals and request rates for three smart contracts

Method	Category	Total	Rate	Valid	Percentage	Avg	p50	p95	p99
registerUser	1000	10	1000	1000	100%	2031.98	2032	2040	2043
		20	1000	1000	100%	2031.58	2031	2039	2042
		50	1000	1000	100%	2030.65	2030	2038	2042
		100	1000	1000	100%	2030.05	2030	2038	2042
		150	1000	1000	100%	2029.72	2030	2038	2040
	2000	10	2000	2000	100%	2032.05	2032	2041	2047
		20	2000	2000	100%	2032.20	2032	2041	2048
		50	2000	2000	100%	2031.10	2031	2040	2046
		100	2000	2000	100%	2030.83	2031	2039	2042
		150	2000	2000	100%	2030.24	2030	2039	2043
	3000	10	3000	3000	100%	2032.47	2032	2041	2048
		20	3000	3000	100%	2032.38	2032	2041	2046
		50	3000	3000	100%	2031.53	2031	2040	2046
		100	3000	3000	100%	2031.12	2031	2040	2043
		150	3000	3000	100%	2030.96	2031	2040	2047
uploadData	1000	10	1000	1000	100%	58.52	58.00	75.00	86.01
		20	1000	1000	100%	66.52	65.00	88.00	102.00
		50	990	990	99%	100.78	97.00	137.00	157.11
		100	990	990	99%	140.00	136.00	185.00	210.11
		150	990	990	99%	181.31	177.00	256.00	298.11
	2000	10	2000	2000	100%	51.00	50.00	63.00	86.01
		20	2000	2000	100%	66.18	62.00	84.05	106.02
		50	1990	1990	99.5%	101.15	100.00	130.00	148.00
		100	2000	2000	100%	133.50	131.00	178.05	232.02
		150	1980	1980	99%	172.02	170.00	228.00	250.21
	3000	10	3000	3000	100%	49.48	49.00	61.00	69.00
		20	3000	3000	100%	63.73	63.00	81.00	90.00
		50	3000	3000	100%	95.58	94.00	123.00	136.00
		100	2980	2980	99.3%	131.32	129.00	169.00	193.21
		150	2990	2990	99.7%	169.54	166.00	223.00	272.11
requestPermission	1000	10	1000	1000	100%	51.51	50.22	66.72	76.93
		20	1000	1000	100%	71.83	70.60	97.11	109.30
		50	1000	1000	100%	101.26	95.54	140.54	216.02
		100	980	980	98%	133.99	131.16	173.95	191.65
		150	980	980	98%	166.48	164.98	219.34	245.67
	2000	10	2000	2000	100%	49.14	48.68	61.33	67.52
		20	2000	2000	100%	63.91	63.29	80.97	87.66
		50	2000	2000	100%	91.75	90.31	117.04	126.64
		100	1990	1990	99.5%	130.25	129.61	162.21	178.44
		150	1990	1990	99.5%	168.20	165.42	228.45	253.96
	3000	10	3000	3000	100%	49.88	49.71	63.74	72.08
		20	3000	3000	100%	71.89	69.24	94.88	107.47
		50	2990	2990	99.7%	92.37	90.91	119.78	134.00
		100	2990	2990	99.7%	132.14	129.62	173.49	196.28
		150	2980	2980	99.3%	168.53	165.85	224.59	249.60

“Avg”, “p50”, “p95”, and “p99” denote end-to-end latency (ms). Percentages indicate success ratio.

Table 3 End-to-end performance of requestView and Confirm under varying totals and request rates

Method	Category	Total	Rate	Valid	Percentage	Avg	p50	p95	p99
requestView		1000	10	1000	100%	57.58	53.00	92.00	142.00
			20	1000	100%	74.00	71.00	108.00	149.00
			50	1000	100%	110.89	107.00	162.00	186.00
			100	1000	100%	147.53	143.00	213.00	242.00
			150	1000	100%	199.48	198.00	261.00	289.00
		2000	10	2000	100%	57.61	56.00	83.00	95.00
			20	2000	100%	72.02	70.00	102.00	118.00
			50	1990	99.5%	1597.96	153.00	237.00	286.00
			100	2000	100%	1657.86	153.00	237.00	286.00
			150	1990	99.5%	1689.72	185.00	271.00	342.00
		3000	10	3000	100%	61.84	61.00	93.00	110.00
			20	2990	99.7%	1098.23	96.00	132.00	159.00
			50	3000	100%	141.40	137.00	208.00	248.00
			100	2990	99.7%	1178.38	173.00	243.00	284.00
			150	2990	99.7%	1178.38	173.00	243.00	284.00
Confirm		1000	10	1000	100%	17.30	17.00	28.00	31.00
			20	1000	100%	24.38	24.00	41.00	45.00
			50	1000	100%	38.39	36.00	65.00	72.00
			100	1000	100%	56.58	54.00	102.00	137.00
			150	1000	100%	73.27	71.00	145.00	179.00
		2000	10	2000	100%	17.57	17.00	30.00	37.00
			20	2000	100%	25.93	24.00	46.00	54.00
			50	2000	100%	38.21	36.00	65.00	72.00
			100	2000	100%	57.63	54.00	100.00	106.00
			150	2000	100%	71.10	67.00	126.00	132.00
		3000	10	3000	100%	16.71	16.00	28.00	31.00
			20	3000	100%	23.26	22.00	38.00	43.00
			50	3000	100%	41.29	40.00	72.00	78.00
			100	3000	100%	55.97	53.00	95.00	106.00
			150	3000	100%	70.74	68.00	117.00	124.00

“Avg”, “p50”, “p95”, and “p99” denote end-to-end latency (ms). Percentages indicate success ratio.

Table 4 End-to-end performance of ViewBundle and ViewAndLog under varying totals and request rates

Method	Category	Total	Rate	Valid	Percentage	Avg	p50	p95	p99
ViewBundle	1000		10	1000	100%	26.89	26.00	48.00	55.00
			20	1000	100%	42.14	40.00	68.00	75.00
			50	1000	100%	62.67	60.00	101.00	112.00
			100	1000	100%	62.67	60.00	101.00	112.00
			150	1000	100%	75.59	74.00	124.00	134.00
	2000		10	2000	100%	19.64	19.00	30.00	34.00
			20	2000	100%	30.83	29.00	53.00	58.00
			50	2000	100%	41.30	39.00	72.00	81.00
			100	2000	100%	79.03	73.00	136.00	153.00
			150	2000	100%	23.21	22.00	38.00	41.00
	3000		20	3000	100%	25.77	25.00	41.00	45.00
			50	3000	100%	41.25	40.00	68.00	73.00
			100	3000	100%	41.25	40.00	68.00	73.00
			150	3000	100%	76.64	74.00	124.00	134.00
ViewAndLog	1000		10	1000	100%	77.52	76.00	107.00	132.00
			20	1000	100%	109.13	106.00	149.00	179.00
			50	1000	100%	153.54	147.00	209.00	266.00
			100	1000	100%	193.54	190.00	266.00	296.00
			150	1000	100%	66.76	64.00	91.00	117.00
	2000		10	2000	100%	76.95	76.00	106.00	117.00
			20	2000	100%	105.67	103.00	146.00	179.00
			50	1990	99.5%	1649.99	147.00	203.00	237.00
			100	2000	100%	61.98	61.00	79.00	95.00
			150	2000	100%	68.51	68.00	87.00	105.00
	3000		20	3000	100%	98.33	97.00	127.00	139.00
			50	2980	99.3%	2142.03	139.00	188.00	208.00
			100	3000	100%	181.96	179.00	236.00	262.00
			150	3000	100%	20.79	20.00	33.00	39.00

“Avg”, “p50”, “p95”, and “p99” denote end-to-end latency (ms). Percentages indicate success ratio.

Table 5 Calculation cost analysis

Scheme	Encryption Calculation	Decryption Calculation
[45]	7E + 3L + 3H	E + H
[46]	4E + 2L + 2H	E + H
Ours	3E + L	2E + L

E, L, and H denote the primitive operations used in each scheme.

Table 6 Time consumption of hash

Aspect	Time				
	1	2	3	4	5
Record	1.22s	1.538s	1.706s	2.451s	3.383s
CID	0.167ms	0.021ms	0.023ms	0.015ms	0.013ms

“Record” and “CID” timings are measured per operation; values are reported in seconds (s) or milliseconds (ms) as shown.

6 Conclusion

This study demonstrates a blockchain-based attribute access control system for safeguarding electronic medical information. This method integrates the blockchain with the IPFS file storage system to provide both on-chain and off-chain storage in an attempt to reduce the storage burden on the chain. It also introduces the idea of multi-attribute authorization center to accomplish fine-grained access, and adopts the idea of hybrid encryption in the processing of *metadata*, which increases the effectiveness of access significantly. However, the visitor's actions is changing dynamically, and there is no guarantee that the visitor will never behave maliciously, so in the subsequent research work, it is intended to introduce the reputation mechanism to strengthen the security of the scheme.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [62502334, 62506255, 62472301, 62372318], National Research Project [2020YFC2006602], Opening Topic Fund of Big Data Intelligent Engineering Laboratory of Jiangsu Province [SDGC2157], Suzhou Blockchain Data Privacy Protection Innovation Application Laboratory, Provincial Key Laboratory for Computer Information Processing Technology, Soochow University [KJS2166].

References

- [1] Kalita, D., Mirza, K.B.: Multivariate glucose forecasting using deep multihead attention layers inside neural basis expansion networks. *IEEE Journal of Biomedical and Health Informatics* **29**(5), 3654–3663 (2025) <https://doi.org/10.1109/JBHI.2025.3530461>
- [2] Ge, L., McInnes, A.N., Widge, A.S., Parhi, K.K.: Prediction of clinical response of transcranial magnetic stimulation treatment for major depressive disorder using hyperdimensional computing. *IEEE Journal of Biomedical and Health Informatics* **29**(5), 3678–3686 (2025) <https://doi.org/10.1109/JBHI.2025.3537757>
- [3] Shi, Y., Wang, M., Liu, H., Zhao, F., Li, A., Chen, X.: Mif: Multi-shot interactive fusion model for cancer survival prediction using pathological image and genomic data. *IEEE Journal of Biomedical and Health Informatics* **29**(5), 3247–3258 (2025) <https://doi.org/10.1109/JBHI.2024.3363161>
- [4] Wu, H., Liu, J., Jiang, T., Zou, Q., Qi, S., Cui, Z., Tiwari, P., Ding, Y.: Attentionmgt-dta: A multi-modal drug-target affinity prediction using graph transformer and attention mechanism. *Neural Netw.* **169**(C), 623–636 (2024) <https://doi.org/10.1016/j.neunet.2023.11.018>
- [5] Liu, J., Guan, S., Zou, Q., Wu, H., Tiwari, P., Ding, Y.: Amdgt: Attention aware multi-modal fusion using a dual graph transformer for drug–disease associations

- prediction. Know.-Based Syst.
- 284**
- (C) (2024)
- <https://doi.org/10.1016/j.knosys.2023.111329>
- [6] Qiu, J., Li, L., Sun, J., Peng, J., Shi, P., Zhang, R., Dong, Y., Lam, K., Lo, F.P.-W., Xiao, B., Yuan, W., Wang, N., Xu, D., Lo, B.: Large ai models in health informatics: Applications, challenges, and the future. IEEE Journal of Biomedical and Health Informatics **27**(12), 6074–6087 (2023) <https://doi.org/10.1109/JBHI.2023.3316750>
 - [7] Liu, J., Hu, F., Zou, Q., Tiwari, P., Wu, H., Ding, Y.: Drug repositioning by multi-aspect heterogeneous graph contrastive learning and positive-fusion negative sampling strategy. Inf. Fusion **112**(C) (2024) <https://doi.org/10.1016/j.inffus.2024.102563>
 - [8] Wu, H., Liu, J., Zhang, R., Lu, Y., Cui, G., Cui, Z., Ding, Y.: A review of deep learning methods for ligand based drug virtual screening. Fundamental Research **4**(4), 715–737 (2024) <https://doi.org/10.1016/j.fmre.2024.02.011>
 - [9] Wang, X.-L.: Model for data access control and sharing based on blockchain. Journal of Software **30**(6), 1661–1669 (2019)
 - [10] Namane, S., Ben Dhaou, I.: Blockchain-based access control techniques for iot applications. Electronics **11**(14) (2022) <https://doi.org/10.3390/electronics11142225> . Article 2225
 - [11] Frank, M., Buhman, J.M., Basin, D.: Role mining with probabilistic models. ACM Trans. Inf. Syst. Secur. **15**(4) (2013) <https://doi.org/10.1145/2445566.2445567>
 - [12] Ferraiolo, D., Cugini, J., Kuhn, D.R., *et al.*: Role-based access control (rbac): Features and motivations. In: Proceedings of 11th Annual Computer Security Application Conference, pp. 241–48 (1995)
 - [13] Lin, A., Brown, R.: The application of security policy to role-based access control and the common data security architecture. Computer Communications **23**(17), 1584–1593 (2000)
 - [14] Park, J.S., Sandhu, R., Ahn, G.-J.: Role-based access control on the web. ACM Transactions on Information and System Security (TISSEC) **4**(1), 37–71 (2001)
 - [15] Barkley, J.: Comparing simple role based access control models and access control lists. In: Proceedings of the Second ACM Workshop on Role-Based Access Control (1997)
 - [16] Xiao, L., Holden, B., Dinghao, W.: Automated synthesis of access control lists. Technical report, College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA (2020)

- [17] Karamnje, A.R., Olanrewaju, O.M., Falalu, S.: Corporate network security using extended access control list (acl) in a simulation environment. *Fudma Journal of Sciences* **3**(3), 264–269 (2019)
- [18] Lindqvist, H.: Mandatory access control. Master’s thesis, Umea University, Department of Computing Science, SE-901 87, Umea, Sweden (2006). Master’s thesis in Computing Science
- [19] Nyanchama, M., Osborn, S.: Modeling mandatory access control in role-based security systems. In: Spooner, D.L., Demurjian, S.A., Dobson, J.E. (eds.) *Database Security IX. IFIP Advances in Information and Communication Technology*. Springer, Boston, MA (1996). https://doi.org/10.1007/978-0-387-34932-9_9
- [20] Li, N., Tripunitara, M.V.: On safety in discretionary access control. In: *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P’05)*. IEEE, ??? (2005)
- [21] Li, N.: Discretionary access control. In: *Encyclopedia of Cryptography and Security*, pp. 353–356. Springer, Boston, MA (2011)
- [22] Hu, V.C., *et al.*: Attribute-based access control. *Computer* **48**(2), 85–88 (2015)
- [23] Servos, D., Osborn, S.L.: Current research and open problems in attribute-based access control. *ACM Computing Surveys (CSUR)* **49**(4), 1–45 (2017)
- [24] Ma, C., Gao, H., Hu, B.: Ciphertext policy attribute-based encryption scheme supporting boolean circuits over ideal lattices. *Journal of Information Security and Applications* **84**, 103822 (2024)
- [25] Suryawanshi, V., Sural, S.: Ciphertext Policy Attribute Based Encryption with Intel SGX. *arXiv preprint* (2024)
- [26] Zhang, Z., *et al.*: Blockchain-based ciphertext-policy attribute-based encryption for controlled multi-user collaboration. *Concurrency and Computation: Practice and Experience* **37**(12–14), 70093 (2025)
- [27] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: *Proceedings of the International Workshop on Public Key Cryptography*. Springer, Berlin, Heidelberg (2011)
- [28] Zhou, Y., *et al.*: Sistema de telemedicina con protección de privacidad basada en cp-abe. *Investigación clínica* **60**(6), 1615–1626 (2019)
- [29] Akinyele, J.A., *et al.*: Securing electronic medical records using attribute-based encryption on mobile devices. In: *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM, ??? (2011)

- [30] Zhu, F., *et al.*: Certificate-based anonymous authentication with efficient aggregation for wireless medical sensor networks. *IEEE Internet of Things Journal* **9**(14), 12299–12218 (2021)
- [31] Chase, M.: Multi-authority attribute based encryption. In: *Theory of Cryptography: 4th Theory of Cryptography Conference (TCC 2007)*, Amsterdam, the Netherlands, February 21–24, 2007, Proceedings 4. Springer, Berlin, Heidelberg (2007)
- [32] Liu, Z., *et al.*: Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In: *Proceedings of the European Symposium on Research in Computer Security*. Springer, Berlin, Heidelberg (2011)
- [33] Ekblaw, A., *et al.*: A case study for blockchain in healthcare: “medrec” prototype for electronic health records and medical research data. In: *Proceedings of the IEEE Open & Big Data Conference*, vol. 13 (2016)
- [34] Zheng, Z., *et al.*: An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems* **105**, 475–491 (2020)
- [35] Ouyang, Z., Shao, J., Zeng, Y.: Pow and pos and related applications. In: *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*. IEEE, ??? (2021)
- [36] Gupta, B.B., *et al.*: Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. *IEEE/CAA Journal of Automatica Sinica* **8**(12), 1877–1890 (2021)
- [37] Chen, B., *et al.*: Bpvse: Publicly verifiable searchable encryption for cloud-assisted electronic health records. *IEEE Transactions on Information Forensics and Security* **18**, 3171–3184 (2023)
- [38] Wang, F., *et al.*: Lightweight and secure data sharing based on proxy re-encryption for blockchain-enabled industrial internet of things. *IEEE Internet of Things Journal* **11**(8), 14115–14126 (2023)
- [39] Jin, B., *et al.*: Bcas: Blockchain-based secure access and sharing scheme for ehr data. *Digital Communications and Networks* (2024)
- [40] Chen, Y., *et al.*: A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection. *Wireless Communications and Mobile Computing* **2021**(1), 6685762 (2021)
- [41] Sukhodolsky, I., Zapechnikov, S.: A blockchain-based access control system for cloud storage. In: *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, ??? (2018)

- [42] Chen, L., *et al.*: Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems* **95**, 420–429 (2019)
- [43] Zhuang, Y., *et al.*: A patient-centric health information exchange framework using blockchain technology. *IEEE Journal of Biomedical and Health Informatics* **24**(8), 2169–2176 (2020)
- [44] Noh, S.-W., *et al.*: Blockchain-based user-centric records management system. *International Journal of Control and Automation* **10**(11), 133–144 (2017)
- [45] Liu, J., Liang, T., Sun, R., Du, X., Guizani, M.: A privacy-preserving medical data sharing scheme based on consortium blockchain. In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6 (2020). <https://doi.org/10.1109/GLOBECOM42002.2020.9348251>
- [46] Cai, C., Liu, Y.: Blockchain-based solution for secure sharing of medical data. *Computer Applications and Software* **41**(12), 360–366 (2024)
- [47] Chen, Y., *et al.*: A bilinear map pairing based authentication scheme for smart grid communications: Pauth. *IEEE Access* **7**, 22633–22643 (2019)
- [48] Kalai, Y.T., Lombardi, A., Vaikuntanathan, V.: SNARGs and PPAD Hardness from the Decisional Diffie-Hellman Assumption. *Cryptology ePrint Archive, Paper 2022/1409* (2022). <https://eprint.iacr.org/2022/1409>
- [49] Servos, D., Osborn, S.L.: Current research and open problems in attribute-based access control. *ACM Computing Surveys (CSUR)* **49**(4), 1–45 (2017)
- [50] Hu, V.C., *et al.*: Attribute-based access control. *Computer* **48**(2), 85–88 (2015)
- [51] Merdassi, I., Ghazel, C., Saidane, L.: A new Itma-abe location and time access security control scheme for mobile cloud. *The Journal of Supercomputing* **79**(11), 12074–12105 (2023)
- [52] Xiong, L., *et al.*: An efficient privacy-preserving access control scheme for cloud computing services. *IEEE Transactions on Consumer Electronics* (2025)
- [53] Beimel, A.: Secret-sharing schemes: A survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Xing, C. (eds.) *Coding and Cryptology, IWCC 2011. Lecture Notes in Computer Science*, vol. 6639, pp. 11–46. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20901-7_2
- [54] Beimel, A., Ben-Efraim, A., Padró, C., Tyomkin, I.: Multi-linear secret-sharing schemes. In: Lindell, Y. (ed.) *Theory of Cryptography. Lecture Notes in Computer Science*, vol. 8349, pp. 394–418. Springer, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_17

- [55] Feneuil, T., Rivain, M.: Threshold Linear Secret Sharing to the Rescue of MPC-in-the-Head. Cryptology ePrint Archive, Paper 2022/1407 (2022). <https://eprint.iacr.org/2022/1407>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [biographiesandphotographsofallauthors.docx](#)