Hindawi
International Journal of Distributed Sensor Networks
Volume 2024, Article ID 5569121, 17 pages
https://doi.org/10.1155/2024/5569121

Research Article

Secure Sharing of Electronic Medical Records Based on Blockchain

Song Luo,[1] N. Han,[1] Tan Hu,[1] and YuHua Qian[2]

[1]*College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China*
[2]*School of Artificial Intelligence, Chongqing University of Technology, Chongqing, China*

Correspondence should be addressed to N. Han; hn1205@stu.cqut.edu.cn

Received 20 June 2023; Revised 14 December 2023; Accepted 8 January 2024; Published 3 February 2024

Academic Editor: Ashish Bagwari

As network technology advances and more people use devices, data storage has become a significant challenge due to the explosive growth of information and the threat of data leaks. In traditional medical institutions, most medical data is stored centrally through cloud computing technology in the institution's data center. This centralized storage method has many security risks, and once the central server is attacked, it will lead to the loss of medical data, which will lead to the leakage of patients' private data. At the same time, electronic medical records are the most critical data in the current medical field. In the traditional centralized healthcare service system (HSS), there are data leakage problems and tampering with electronic medical records due to human factors. At the same time, each hospital is built independently, resulting in the current centralized healthcare service system having a data silo problem, making it difficult to share medical data between institutions securely. With the increase in the number of users in the system, the electronic medical record data in the system also increases gradually, resulting in the increasing overhead of decryption calculation. Therefore, this paper proposes a blockchain-based access control scheme with multiparty authorization to ensure the security of electronic medical records. The scheme uses an SM encryption algorithm to encrypt the medical data in the system. It adds the patient's signature to ensure the confidentiality and security of the data, and the encrypted electronic medical records (EMRs) are stored in the InterPlanetary File System (IPFS) to realize the distributed storage of EMR. In addition, role-based multiauthorization access control is implemented through smart contract-based to ensure the security of EMR. We have analyzed the security of this paper's solution and compared its performance with the existing schemes based on other cryptographic algorithms. The experimental results show that the proposed solution significantly improves the secure sharing of EMR and provides system performance.

1. Introduction

The widespread adoption of Internet technology has led to significant advancements in healthcare information technology. Electronic medical records (EMRs) have been extensively used within hospitals' healthcare service systems (HSS) to assist medical practitioners in diagnosing and treating patients. However, due to the substantial volume and sensitive nature of EMR, healthcare institutions are vulnerable to cybersecurity attacks. The COVID-19 pandemic has witnessed several incidents of sensitive data breaches in the healthcare industry, highlighting the importance of security technologies related to EMR [1].

With the rapid growth of EMR data, when a patient visits a hospital or healthcare facility, a corresponding EMR is generated between the hospital and the patient and stored in the system database. For patients, the previously developed EMRs are also needed when they visit other hospitals. The relevant EMR of the patient may be stored in different hospitals. The doctor needs the patient's medical data for the next treatment step. In this case, the doctor is unaware of the patient's previous medical history, which leads to easy misdiagnosis, and this is when the problem of data silos in hospitals becomes quite prominent [2]. For hospitals and medical institutions, hospitals or medical institutions need to have corresponding standards or regulations

for exchanging and transferring data when exchanging and transferring medical data [3]. Currently, most EMRs are stored centrally; i.e., the medical data are stored on a centralized cloud server. However, this storage method can cause data loss or tampering and lead to a single-point failure of the entire system, such as a natural disaster or hacking of the cloud server. In addition, the cloud server is always a semitrusted entity. If it is attacked and returns incorrect or incomplete medical record data, it may lead to misjudgment by users (e.g., doctors and healthcare organizations), which may endanger patients' lives. Finally, the need to protect healthcare data from security breaches and criminal activities has become increasingly urgent; if someone not authorized by the system obtains sensitive patient data, then this data can be sold or redistributed on the open market, resulting in the exposure of sensitive patient information to anyone, and when electronic healthcare data is distributed among different healthcare organizations in an available network, data tampering, and forgery, among other attacks can quickly occur [4]. Therefore, the problem of secure sharing of EMR increases the cost of healthcare services for patients and affects effective communication and cooperation among healthcare organizations. Thus, to address these problems, if we design a secure and reliable decentralized electronic medical data-sharing scheme that provides accurate, secure, and timely patient-sensitive data and enables secure sharing of EMR across institutions, it can accelerate the research on medicine and diseases.

With the development of blockchain technology in recent years, it has been widely used in various industries due to its characteristics of invariance, data integrity, and distributed storage, so many scholars have begun to put forward many solutions to the current problems of EMR [5]. Literature [6–8] uses blockchain architecture to implement electronic medical record storage solutions. Still, these solutions only store part of the data on the blockchain and do not solve the problem of centralized nodes in the current system. Literature [9], for the first time, combined blockchain and InterPlanetary File System (IPFS) to propose a new model of electronic medical record scheme. Still, this author did not consider the problem of access control centralization, which led to the increased computational overhead of the EMR stored in the system. Literature [10] and literature [11], based on literature [9], used attribute-based encryption (CP-ABE) to allow doctors and patients to formulate access policies for EMR, which realized fine-grained access control for data requesters. Still, this access control scheme increased the computational overhead of the system's access control. In addition, some researchers have adopted a semidecentralized architecture to improve the response speed of the whole system; i.e., the system behavior of storing EMR data is outsourced to external trusted cloud storage servers, such as in literature [12] and literature [13]. Still, this approach does not avoid the attacks during the data transmission process or the attacks on the external cloud storage servers. Based on this problem, some researchers have proposed using fog computing to avoid attackers' attacks during data transmission and on cloud servers, such as in literature [14]. Still, literature [14] does not achieve user and attribute revocation, which enables access control to the system.

This paper proposes a new electronic medical record scheme that uses blockchain and IPFS technology to address the existing problems. In this scheme, the hash address of EMR data stored in IPFS is uploaded to the blockchain after encryption, ensuring the security of the EMR data stored in IPFS. The decentralized feature of IPFS avoids the single point of failure problem of the entire system. Meanwhile, we also introduce the SM2 signature algorithm to support the authentication of EMR data, which avoids the data leakage problem during data transmission. In addition, this paper designs an EMR data-sharing strategy through which it can balance the data transactions of EMR in the whole system and indirectly promote the EMR data transactions in the system. Finally, this scheme proposes a multiauthorization role control scheme based on smart contracts, which achieves lightweight access control and dramatically reduces the overall access control overhead in the system compared to the traditional role control scheme. The following are the main contributions of this paper:

(1) We propose integrating blockchain technology and IPFS technology to create a decentralized solution for electronic medical records. This solution ensures distributed data privacy protection for users within the system. By leveraging IPFS technology, encrypted medical data can be securely submitted, thereby reducing the overall storage overhead of the system. Furthermore, the solution records the corresponding storage address and medical-related information on the blockchain, guaranteeing effective data storage and preventing any tampering with the data

(2) We suggest using a hybrid encryption method of SM2 signature algorithm and SM4 encryption algorithm to ensure secure EMR data transmission

(3) We have developed an EMR data pricing strategy that calculates the potential value of EMR data. This strategy facilitates the sharing of EMR data across different agencies and ensures a balanced exchange of EMR data between data providers and consumers within the system

(4) We propose using smart contract technology to implement a smart contract-based multiauthorization role access control system. This approach is aimed at streamlining access control within the system, reducing the overall access control overhead

2. Related Work

2.1. Blockchain-Based Electronic Medical Record Model. In recent years, blockchain technology has experienced rapid development and is characterized by its decentralized nature, data immutability, and transparency. It was first proposed by Nakamoto [15] in 2008 and initially applied in the context of Bitcoin. Blockchain technology can be seen as a distributed database that enables reliable and secure data

storage within the system, eliminating the need for third-party cloud storage and addressing potential data security issues associated with cloud storage. Building upon these distinctive features, researchers have begun exploring the application of blockchain technology in the medical field. Azaria et al. [6] introduced MedRec, an electronic medical record model based on blockchain technology, which enhances the security of data transmission. However, this model still relies on centralized cloud servers, making it susceptible to data leakage and malicious tampering. Liang et al. [7] proposed a secure data transfer scheme using Fabric blockchain, which improves data transfer security and reduces communication overhead. Zhang and Poslad [8] put forth a fine-grained access control-based secure storage model for electronic medical records, enabling granular access control by authorizing different queries. This blockchain-based approach effectively addresses data requests without revealing unauthorized personal information. However, it increases the computational consumption of querying electronic medical records, resulting in inefficiencies for the entire system. Nevertheless, the calculation of tokens is susceptible to attacks or tampering. Sun et al. [9] ensured the security of the storage platform by suggesting decentralized storage of encrypted electronic medical data in IPFS. Only the hash address returned from IPFS is uploaded to the blockchain. Li et al. [16] proposed a transaction record-centric system and utilized game theory to introduce a new token economic system, aiming to optimize the process of sharing electronic medical records and achieve a Nash equilibrium point [17]. Liu et al. [18] combined searchable encryption with federated blockchain to achieve efficient and reliable multikeyword searches. However, there is still a presence of cloud servers in the system architecture. However, the transaction record-centric approach makes the system inefficient and prone to congestion. Jayabalan and Jeyanthi [19] addressed the system overhead by adopting a patient-centric model and utilizing IPFS. However, handing over the entire system network maintenance to the patient is challenging to implement in reality, and centralized access control remains an issue. Kaur et al. [11] proposed a blockchain-based approach for EMR storage and sharing by utilizing IPFS as an off-chain storage and encrypting the patient-related records while encrypting and decrypting them using the CP-ABE algorithm. However, it still suffers from a high system overhead and centralization of the access control. Alrebdi et al. [20] proposed a blockchain-based scheme for secure storage and access to EMRs through a combination of IPFS and cloud storage to achieve search and verification of encrypted files. Still, the scheme does not achieve better access control, while the existence of cloud storage leads to system vulnerability to attacks. Ramesh et al. [21] proposed blockchain-based tamper-proof EMR storage access in cloud environment. Still, the third-party reviewer in the scheme is susceptible to attacks due to the lack of a distributed architecture, which leads to the review process of a single point of failure. Mohammed et al. [22] proposed a blockchain-based distributed EMR system, which achieves secure storage of EMR data through biological signaling and light-weighted encryption, but did not design a corresponding access control policy. Table 1 demonstrates a comparative analysis of existing EMR models and proposals.

2.2. Smart Contract-Based Access Control Technology. The concept of smart contracts traces its origins back to the paper by Szabo, which introduced the idea of using computer code and cryptographic techniques to automate contract execution and ensure the immutability and nonrepudiation of their execution [27]. With the advancement of blockchain technology, smart contracts have gained widespread adoption and implementation. They are designed as code or programs embedded within the blockchain that can autonomously execute and update its state. The emergence of smart contract platforms like Ethereum has propelled the development of smart contract technology. Ethereum introduced Solidity, a Turing-complete language for smart contracts, and provided developers with tools and environments to create and deploy smart contracts [28]. Building upon these features, researchers have explored the use of smart contracts for access control. For instance, Cruz et al. [23] implemented a role-based access control mechanism using smart contract technology, enabling cross-organizational role usage. Zhang et al. [24] proposed a framework for IoT environments by introducing multiple access control contracts to achieve enhanced access control. In access control, Wang et al. [25] utilized smart contract technology to implement access control and revocation within the Internet of Things (IoT) environment using a single contract. However, they did not tackle the security vulnerabilities of a lone access control node. Maesa et al. [26] introduced a decentralized and dynamic access control approach based on policies, allowing for complex access control policies based on user or environmental properties. These studies highlight the potential of smart contract technology in implementing robust and flexible access control mechanisms within various domains. Table 2 demonstrates the comparative analysis of existing blockchain-based access control techniques with the proposal.

3. Preliminaries

3.1. Blockchain. In recent years, blockchain technology has become prevalent in developing distributed systems. Blockchain is a data structure that connects each block through hash pointers in chronological order, resulting in an immutable distributed ledger combined with cryptographic techniques [29]. Each block within the blockchain comprises at least five essential parameters: timestamp, the hash of the previous block, nonce, target hash, and the Merkle root of all transaction records. The block body encompasses all the transaction records.

Blockchain can be classified into three main types: public chains, consortium chains, and private chains [30]. Public chains are open and accessible to anyone without any access restrictions, enabling participation in transactions on the chain by any individual. Consortium chains are blockchain networks maintained collectively by institutions or organizations. Participants in the consortium chain possess management permissions and decision-making authority over the chain. To join the consortium chain, individuals must undergo a qualification review. Once their qualifications

Table 1: Comparison of existing blockchain-based EMR schemes with the proposed schemes.

Schemes	Privacy protection	Access control	Blockchain	IPFS	Incentives
Azaria et al. [6]	x	✓	✓	x	x
Liang et al. [7]	x	✓	✓	x	x
Zhang and Poslad [8]	✓	✓	✓	x	x
Sun et al. [9]	✓	x	✓	✓	x
Li et al. [16]	✓	✓	✓	x	✓
Sun et al. [10]	✓	✓	✓	✓	x
Liu et al. [18]	✓	✓	✓	x	x
Jayabalan and Jeyanthi [19]	✓	✓	✓	x	x
Kaur et al. [11]	✓	✓	✓	✓	x
Alrebdi et al. [20]	✓	✓	✓	✓	x
Ramesh et al. [21]	✓	✓	✓	✓	x
Mohammed et al. [22]	✓	✓	✓	x	x
Ours	✓	✓	✓	✓	✓

Table 2: Comparison of existing smart contract-based access control schemes with the proposed scheme.

Schemes	Smart contract	Multiauthorization	Access revoked
Cruz et al. [23]	✓	x	x
Zhang et al. [24]	✓	✓	x
Wang et al. [25]	✓	x	✓
Maesa et al. [26]	✓	✓	x
Ours	✓	✓	✓

are approved, they are granted access to the chain. Private chains, on the other hand, are maintained and managed by a single organization or individual, offering complete control over the entire blockchain to the owner.

3.2. Ethereum (ETH). In 2013, Buterin [31] drew inspiration from Bitcoin and introduced Ethereum (ETH). Ethereum is a blockchain platform that enables individuals to construct and utilize decentralized applications. The Ethereum platform offers an Ethereum virtual machine (EVM), a "Turing-complete" virtual machine. Users can leverage the EVM to develop and execute smart contracts.

Smart contracts are software programs that operate on the Ethereum virtual machine (EVM) and possess their own Ethereum accounts. Upon receiving transaction data, they autonomously execute predetermined code logic. Smart contracts are capable of invoking other smart contracts as well. They facilitate the automated execution of contracts between parties, with the entire process being recorded on the trusted public ledger without individual interference. Additionally, smart contracts can store data and trigger events based on predefined conditions.

3.3. InterPlanetary File System (IPFS). The InterPlanetary File System (IPFS) is a decentralized file system and peer-to-peer hypermedia protocol. Benet [32] recognized the limitations of data distribution in traditional HTTP protocols and proposed the InterPlanetary File System (IPFS) by incorporating some aspects from the BitTorrent content distribution protocol. In IPFS, storage is distributed among nodes within the IPFS network, ensuring scalability. Each node is assigned a unique node ID, which corresponds to the hash value of its public key. Every node maintains a distributed hash table (DHT) to retrieve network addresses of other nodes. The DHT fulfills specific functions and facilitates node discovery within the network. When a file is uploaded to IPFS, the system generates a unique hash value, allowing a single instance of the file to be stored in IPFS.

3.4. SM2 Signature Algorithm. The SM2 algorithm is an asymmetric encryption algorithm introduced by the China National Cryptography Administration in 2010 [33]. It is aimed at replacing the widely employed 1024-bit RSA algorithm. In comparison to RSA, the SM2 algorithm exhibits increased computational complexity but faster encryption and decryption speeds, leading to reduced resource consumption on devices.

The process description of the SM2 signature algorithm is as follows:

(1) $M' = Z_A \| M$, where M represents the data to be signed and Z_A denotes the identifiable information of A, partial elliptic curve system parameters, and the hash value of user A public key

(2) Compute $e = \mathrm{Hash}(M')$ and convert it to an integer

(3) Generate a random number $k \in [1, n-1]$ using a random number generator

(4) Compute the elliptic curve point $(x_1, y_2) = [k]G$, and convert it to an integer. Here, G represents a point on the elliptic curve, and $[k]G$ denotes the multiplication of the point by a scalar

(5) Compute $r = (e + x_1) \bmod n$. If $r = 0$ or $r + k = n$, return to (3).

(6) Compute $s = [(1 + d_A)^{-1}(k - r * d_A)] \bmod n$. If $s = 0$, return to (3) to regenerate a random number

(7) Convert r and s to strings to obtain the signature value (r, s) for the message

3.5. SM4 Encryption Algorithm. The SM4 algorithm operates on 128-bit plaintext and ciphertext using a 128-bit key. Both encryption and decryption employ the same key [34]. The encryption algorithm, including key expansion, is based on a 32-round nonlinear iterative function. The core of the data encryption process involves a round function that combines linear and nonlinear operations. Initially, the key is divided into four groups of 32 bits each, and the key expansion algorithm generates 32 groups of 32-bit keys. The 128-bit input data is processed iteratively, with each round handling four groups of 32 bits. The overall structure of the SM4 symmetric encryption algorithm is illustrated in Figure 1.

3.6. Role-Based Access Control. The role-based access control (RBAC) model is a contemporary model introduced in the late 20th century [35]. In RBAC, access permissions are represented by three key components (who, what, and how). These components define the following: who can perform how on what, and the evaluation process determines the truth value of the logical expression. Who represents the owner or subject of the permission, such as users and roles. What refers to a resource or object, while how means an operation or activity. RBAC decomposes all permissions into subsets and defines them as corresponding roles, which are then assigned to respective subjects. The fundamental elements of the issue include users, roles, sessions, and permissions. Various forms of RBAC models can be constructed, including RBAC0 (core RBAC), RBAC1 (hierarchical RBAC), RBAC2 (constraint RBAC), and RBAC3 (combined RBAC) [36].

4. Proposed Model

This section provides a detailed description of the system design. Patients, as creators of EMR data, upload their encrypted data to the system. Medical institutions act as IPFS nodes, responsible for storing the encrypted EMR data and recording its hash address on the blockchain. This section presents the system model, outlining the secure transmission of data and access control mechanisms. Finally, the chapter discusses the pricing strategy for EMR data.

4.1. System Framework. As shown in Figure 2, traditional healthcare systems are centralized, and as the number of users increases, so do the system costs. They are susceptible to single points of failure and data breaches. In contrast, this paper presents a blockchain-based system where medical institutions are network nodes to maintain the system. EMR data generated by patients is uploaded to the IPFS network for storage, and with patient authorization, it can be shared among medical institutions. This distributed system enhances security and reliability, mitigating the risk of single points of failure.

As shown in Figure 3, the system consists of entities such as patients, doctors, IPFS, and the institution-maintained blockchain. The MedCoin token system, in conjunction with incentive mechanisms, regulates the sharing process of EMR (electronic medical record) data within the system and rewards users for sharing their EMR data. The entities involved in the system are listed in Table 3, with the primary commodities being EMR, patients, doctors, IPFS, and the healthcare institution-maintained blockchain.

4.2. System Entities. The primary system entities include EMR, patients, doctors, IPFS, and the blockchain, which healthcare institutions maintain. These entities are depicted in Table 3.

4.2.1. EMR. When a patient receives medical treatment, the system records the entire process. Initially, the patient enters the system to initialize their EMR. Subsequently, the doctor completes the medical history and prescription within the EMR. Once the entries are finalized, the EMR is encrypted and transmitted to the IPFS node network, maintained by healthcare institutions to ensure data confidentiality and prevent third-party access. The IPFS node returns a hash address, which the system encrypts and uploads onto the healthcare institution's blockchain. This information is encapsulated into a block and added to the blockchain following a consensus process.

4.2.2. Patients. Patients play a vital role in the system as they are the primary creators of EMRs and have complete control over them. By creating EMRs, patients can receive medical treatment across different healthcare institutions, greatly enhancing their accessibility to care.

4.2.3. Doctors. Doctors serve as secondary creators of EMR data and have the authority to access EMR records authorized by patients. After patients initialize their EMR, doctors record medical visit data, enabling it to become a complete EMR and circulate within the system.

4.2.4. Medical Institution. The healthcare institution is the creator and maintainer of the system and is responsible for user privacy and EMR data security. It constitutes the distributed storage network IPFS, which stores EMR data. In the incentive mechanism, the healthcare organization acts as an intermediary that responds to data consumption requests, obtains MedCoin tokens, and transfers them to the EMR data creator.

4.2.5. MedCoin. MedCoin tokens were introduced to encourage the sharing of EMR data. When EMR data is shared, the system responds to the consumer's request, transmits the data to them, and receives MedCoin as payment, which is transferred to the data creator to reward the data-sharing behavior. In this way, complete EMR data sharing is realized through the amount of MedCoin. The specific structure of MedCoin is shown in Figure 4.

4.3. Secure Storage and Sharing of EMR Data. EMR data comprises sensitive information, including patient personal details and doctor diagnostic records. The system must ensure secure storage and sharing of EMR data while protecting patient privacy. In some existing solutions, such as

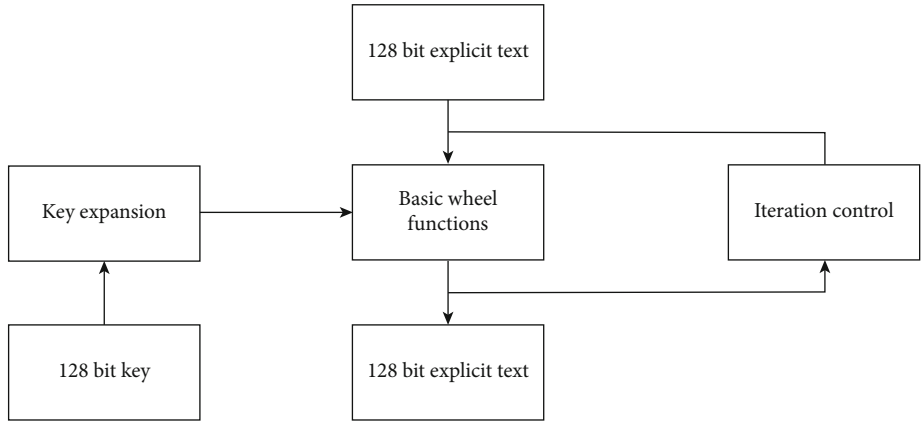

Figure 1: Overall architecture of SM4.

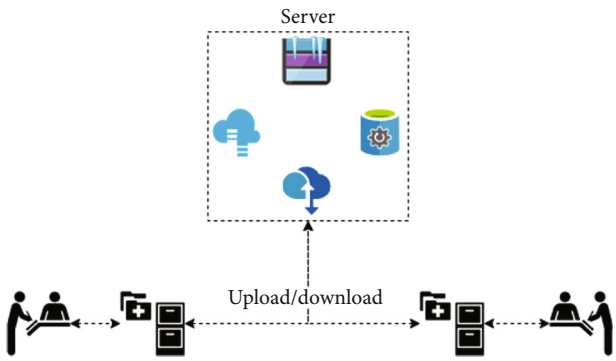

Figure 2: Centralized traditional healthcare system.

those described in literature [10, 11], CP-ABE is employed to address the secure storage of EMR data. However, this paper utilizes a more lightweight SM encryption algorithm to encrypt EMR data, aiming to resolve the security concerns associated with storing and sharing EMR data within the system.

4.3.1. Secure Storage of EMR Data. With distributed storage technology, this paper realizes the secure sharing of EMR data among various medical institutions through blockchain technology. The fundamental operations of the storage of EMR data within the system are outlined as follows:

Initialization: when patients access the system, they undergo an EMR initialization process. Here, patients supply their essential personal information and complete relevant EMR fields.

Generate patient's public-private key pair: the system generates a patient's key pair using their ID, creating a private-public key pair based on the provided initial parameters γ and patient ID:

$$(\text{private\_key}, \text{public\_key}) = \text{KeyGenerator}(\gamma, \text{ID}). \quad (1)$$

Encryption and generation of signature: the system conducts a hash operation on unencrypted EMR data, computing its digest value. Using the patient's private key private_key and the computed digest value, the system employs the SM2 signature algorithm to generate the EMR data's signature. Furthermore, the system utilizes the SM4 encryption algorithm to calculate CT_{emr}, concatenated with Sign to produce the patient's EMR data ciphertext CT, as shown in Algorithm 1.

Upload IPFS network: upon receiving the encrypted EMR data, the system follows a secure storage process. Initially, the encrypted patient EMR is uploaded to the IPFS network. The hash address provided by the IPFS network, along with the patient's ID and relevant data, is subsequently uploaded to the system's blockchain as a package. The system awaits authentication by the chain's nodes. Upon successful mutual authentication, the package is compiled into a block and added to the system's blockchain. Additionally, the system compiles the patient's ID, public_key, and signature Sign into a list and uploads it to the IPFS network, obtaining a corresponding hash address. After confirmation by the chain's nodes using the blockchain's broadcast mechanism, this value is packaged and uploaded to the chain. Simultaneously, the signature Sign generated by the patient during the encryption phase is automatically packaged and uploaded to the blockchain by the system for patient EMR data verification.

4.3.2. Secure Sharing of EMR Data. Once the patient's EMR data has undergone the steps illustrated in Figure 5, which include steps ① and ②, the data is securely stored on the IPFS within the system. Subsequently, through the steps depicted as ③, ④, and ⑤ in Figure 5, the hash address and the patient's ID returned by the IPFS are bundled and uploaded to the blockchain. When a physician or institution initiates a request to access a patient's EMR, the following detailed description explains how the data is securely shared within the system.

Initiating a request: a third-party user, such as a physician or institution, initiates an EMR data access request within the system. Upon receiving this request, the system seeks authorization from the patient who owns the EMR data. The system waits for the patient's approval, as shown in Figure 5, during steps ⑦ and ⑧.

Data lookup: upon receiving remote patient authorization for a data request initiated by a third-party organization or doctor, the organization's node will execute the deployed

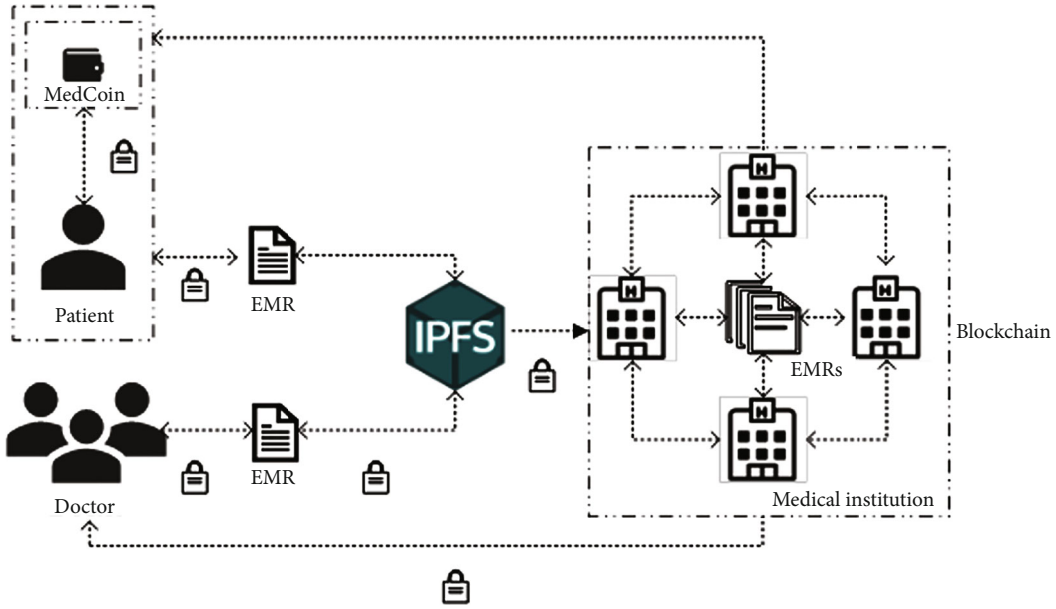

Figure 3: Proposed system framework.

Table 3: Entities in the system.

System entity	Description
Patients	Creators and owners of EMR data
Doctors	Secondary creators of EMR
MedCoin token wallet	Deposit of MedCoin tokens awarded by the system
IPFS	Distributed storage system to store encrypted EMR
Medical institutions	Maintainer of the system blockchain network
EMR	Electronic medical records

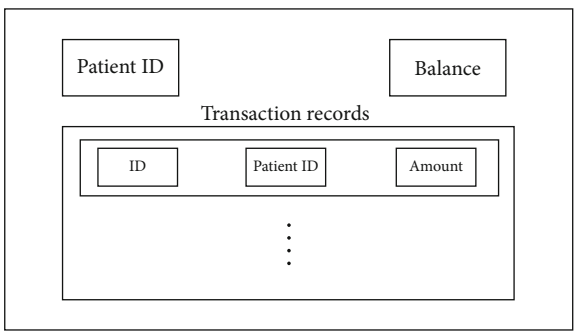

Figure 4: MedCoin storage structure.

smart contract on the blockchain. This contract will conduct a lookup using the patient's ID and relevant keywords, retrieving the hash value from the packaged IPFS and the patient's blockchain signature. This process is depicted in Figure 5, particularly in steps ⑧-⑪.

Signature verification: when a patient remotely authorizes a data request from a third-party organization or doctor, the system looks up the hash value in the blockchain using the patient's ID and keywords. It then retrieves the patient's public_key and signature Sign from IPFS based on the obtained hash value. Subsequently, the system compares the signature found by the third-party organization with the system's Sign2 to determine if they match. If they match, the process proceeds; if they differ, the system denies further access to the third-party organization, as in steps ⑫-⑭ in Figure 5.

$$\text{CompareSign} = \text{Sign1} \oplus \text{Sign2}. \tag{2}$$

Data decryption and signature verification: if the signature comparison succeeds, the system decrypts and verifies the data. It retrieves the EMR data cipher stored in IPFS based on the hash value provided by the third-party organization or doctor. The system then decrypts and verifies the signature. It calculates the digest value of the current data. It uses the SM2 signature verification algorithm, taking into account the signature Sign obtained from the third-party organization or doctor, the patient's public_key, and the digest value H_1 of the data. If the verification fails, the system rejects the operation. If the verification is successful, the system decrypts the EMR data and provides the plaintext M_{emr} to the third-party institution or physician, as shown in Figure 5, step ⑮. During the EMR data decryption stage, the third-party organization cannot access the patient's public-private key pair and can only retrieve the patient's signature stored in the blockchain. This approach ensures the security of patient data by conducting encryption, decryption, and verification operations through a system trusted by the patient, as shown in Algorithm 2.

4.4. A Sharing Strategy Based on Assessing the Value of EMR Data. Existing healthcare data security storage solutions face challenges because patients, physicians, and healthcare

Input: Medical record data M_{emr}, Patient ID
Output: $CT = CT_{emr} \| Sign$
1: KeyGen: patient's $private\_key$ and $public\_key$
2: Extract:$H_1 = SM3(ID \| public\_key \| a \| b \| x_G \| y_g)$
3: Calculating hashes:$H = SM3(H_1 \| M_{emr})$
4: Generate signature: $Sign = SM2(H \| private_{key})$
5: Encrypt: $CT_{emr} = SM4(M_{emr})$

Algorithm 1: EMR data signature and encryption algorithm based on SM2 algorithm.

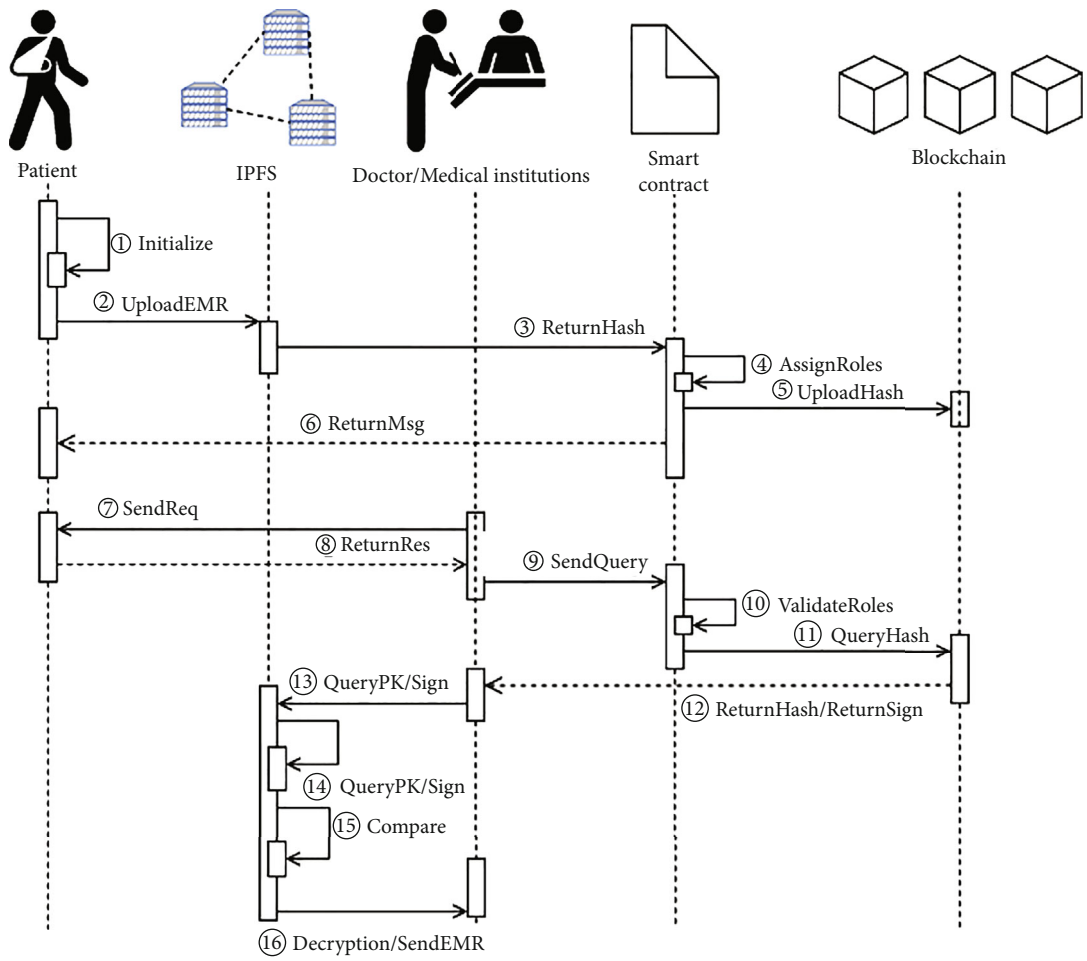

Figure 5: Data flow diagram of system entities.

organizations operate independently in the system, often lacking motivation for active EMR data sharing [37]. To tackle this problem, this paper introduces an analytical strategy to assess the value of EMR data, aiming to overcome obstacles in sharing EMR data among diverse entities.

4.4.1. Entities in the Sharing Strategy

(1) Data Provider. In the proposed sharing strategy, the data provider is the patient who created the EMR data within the system. The patient maintains full control over their EMR and acts as the data provider, enabling them to grant licenses for sharing their self-generated EMR data with others.

(2) Data Consumer. In sharing strategies, data consumers are typically doctors or institutions seeking access to EMR data shared by data providers in the system. Data consumers are not the original creators of the EMR data. When they receive EMR data from a specific data provider within the system, the system rewards the data provider with MedCoin tokens as an incentive for sharing their EMR data.

Input: Encrypted ciphertext CT_{emr}
Output: Medical Record Data M_{emr}
1: **if** Sign verification successed **then**
2: $H_1 \longleftarrow Extract(ID, public\_key, a, b, x_G, y_G)$
3: $Bool \longleftarrow SM2VerifySign(H_1, Sign, public\_key)$
4: **if** $Bool == True$ **then**
5: Verification Success
6: **return** $M_{emr} \longleftarrow SM4(CT_{emr})$
7: **else**
8: System Refused
9: **return** False
10: **end if**
11: **else**
12: **return** False
13: **end if**

Algorithm 2: EMR data decryption and signature verification algorithm based on SM2 algorithm.

(3) Leader. In the sharing strategy, the leader, usually a medical institution, takes a proactive role. It evaluates the value of EMR data and sets the reference price, initiating the bidding process with the first bid. Data consumers can assess the data's value based on the leader's bid and place their bids accordingly.

4.4.2. Security Price Algorithm Based on Evaluating the Value of EMR. In this paper, we enhance the optimal EMR price algorithm proposed by Li et al. [16] by incorporating the calculation of potential weight values for each EMR transaction. This is achieved by evaluating the potential value of the EMR. The security price algorithm, which evaluates the value of the EMR, is described as follows:

Initialize algorithm parameters, including the initial evaluation price p of EMR, data consumer bid price d, system maintenance consumption c, and other relevant parameters.

Assess the EMR's potential value through an evaluation algorithm that analyzes requested EMR data and calculates its anticipated value based on the patient's EMR data when a data consumer sends a request.

$$D_i = \ln(j \cdot n + k), \tag{3}$$

where D_i represents the weight of each EMR calculated, j represents the disease weight, n represents the number of diseases, and k represents the nonzero coefficient.

$$P_v = i \cdot D_i + c, \tag{4}$$

where P_v represents the calculated minimum prognosis for each EMR, D_i represents the weight of the i-th EMR, and c represents the consumption of the maintenance system.

The leader sets the EMR price bidding strategy. The leader in the system is the system itself, and the system sets the EMR price bidding strategy based on the expected estimate of EMR calculated in the previous step as

$$P = \{P_i, P_v \le P_i \le P_{\max}\} i \in N. \tag{5}$$

Data consumers calculate the optimal price for current EMR data:

$$d_i = \frac{\alpha_i}{P_i} + P_v - c, \tag{6}$$

where α_i is the predefined nonzero positive factor and d_i represents the best price of the i-th EMR.

The data provider calculates the optimal return. The data provider calculates the return on its own EMR as

$$R_w = d_i - P_i - c + \gamma, \tag{7}$$

where γ is the predefined nonzero positive factor and R_w represents the optimal return price of the owner EMR data.

The system will pay the corresponding MedCoin tokens to the data provider according to the calculated price; MedCoin will record the corresponding data consumption records, the best price of the data consumer, and the best revenue of the data provider; and the MedCoin payment records will be encrypted by the system and then packaged and uploaded to the chain by the system, as shown in Algorithm 3.

4.5. Role-Based Access Control Based on Smart Contracts and Multiple Authorized Organizations. Literature [6–9, 18] centralized system access control, posing security risks due to a central server storing extensive access information, making it an attractive target for potential attackers. Breaches could grant unauthorized access to all system resources, potentially resulting in data breaches or misuse. To address these challenges, this paper introduces an extended role-based access control scheme called multiauthority role-based access control for smart contracts (MARBAC-SC), which utilizes smart contracts and multiauthority mechanisms. MARBAC-SC enhances the traditional RBAC model and is depicted in Figure 6 for reference, and the main flow is shown in Figure 7.

Contract deployment: after the system is initialized, the organization node in the system executes the organization

Input: Initial Parameters p, d, c, j, n, γ
Output: the unit value and reward R_w^*, d_i^*

```
if Data Provider passed request then
  for each EMR do
    j ⟵ Sum_Num()
    D_i ⟵ computeEMRWeight(n, j)
    P_v ⟵ computeEMRValue(D_i, c)
  end for
  for each Data Consumer i do
    for the unit price P set value from P_v to P_max by leader do
      if d_i<c then
        break
      end if
      data consumer i computing value depend on:
      d_i = (α_i/P_i) + P_v − c
      data provider computing value depends on:
      the unit value and reward R_w^* = R_w, d_i^* = d_i
    end for
  end for
else
  return
end if
```

Algorithm 3: Price algorithm based on evaluating the value of EMR.

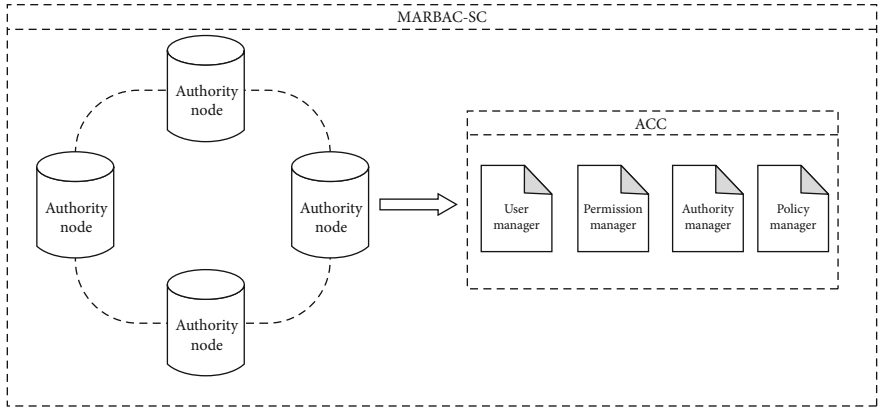

Figure 6: The architecture of MARBAC-SC.

essential generation function, generates the organization's corresponding public-private key pair O_public_key and O_private_key, and executes the deployment of the smart contract to the chain.

Add users and assign corresponding roles: after the contract is deployed and the user fills in the personal information, the system will automatically execute the AddUser function in the contract to add the user to the corresponding organization after the user clicks to create the EMR for the individual, as shown in Algorithm 4.

Remove authenticated users: if a user has not been active for a long time, an organization in the system can execute the removeUser function to remove the role assigned to the current user. After the function is completed, the role of the user recorded in the chain will be removed, as shown in Algorithm 5.

Role access policy determination: after the user's role is assigned, if the user performs a function in the system that involves permission control, the system will execute an access policy judgment in advance to determine whether the current user has permission to call the method, as shown in Algorithm 6.

5. Solution Analysis

Unlike traditional medical visit systems, this solution utilizes blockchain and the IPFS storage network to shift from a centralized to a decentralized system. This transformation allows patients to access their EMR data across different institutions. The blockchain and IPFS storage network, maintained by healthcare institutions, guarantee secure and efficient storage of patient EMR data, safeguarding sensitive

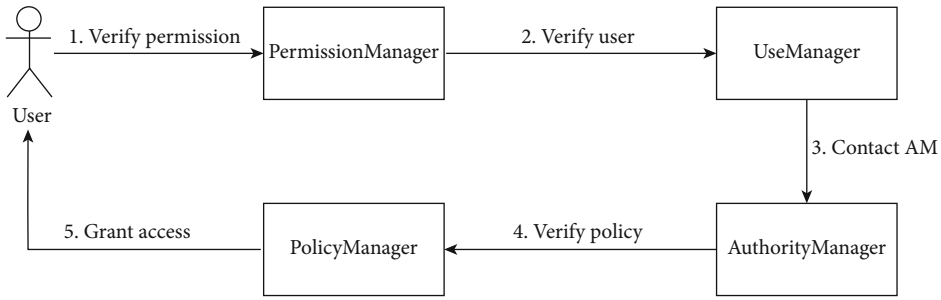

Figure 7: Main processes of MARBAC-SC.

Input: *O_public_key*,*userAddress*
Output: grantRole(Role, userAddress)
1: **if** *status* == *True* **then**
2: **if** userAll[userAddress].userSince==0 **then**
3: *userAll*[*userAddress*] = *User*(*O_PublicKey*, *block.timestamp*)
4: userAccounts.push(userAddress)
5: indexUser++
6: grantRole(Role, userAddress)
7: **else**
8: **return**
9: **end if**
10: **else**
11: **return**
12: **end if**

Algorithm 4: Add users and assign corresponding roles.

Input: *O_public_key*,*userAddress*
Output: revokeRole(Role,userAddress)
1: **if** userAll[userAddress].userSince != 0 **then**
2: delete userAccouts[userAll[userAddress].userIndex]
3: userAccouts.pop()
4: delete userAll[userAddress]
5: indexUser - -
6: revokeRole(Role,userAddress)
7: **else**
8: **return** "User does not exist"
9: **end if**

Algorithm 5: Remove authenticated users.

patient information. In this section, our principal emphasis lies in the execution of experimental analyses and scenario comparisons, aimed at assessing the efficacy of the proposed scheme.

5.1. System Deployment. All simulations of this program are executed on an Intel Core i7 processor with 16 GB of RAM to validate our framework. Smart contracts in Ethereum are written by Solidity, a programming language through which one can make transactions on the chain. This solution utilizes the web3.js framework to access the change account via HTTP connection in JsonRPC, compiling the smart contract requires the use of Ganache, and the Truffle development environment must also be used.

5.2. Security Analysis. Ensuring the security of patient data, especially within EMR, is paramount. Preserving EMR confidentiality and integrity is vital for protecting sensitive patient information and preventing unauthorized access or data breaches. In this context, the security level offered by this solution is demonstrated through the following aspects:

Eliminating single points of failure: the proposed scheme in this paper employs decentralized blockchain technology to effectively reduce the risk of system-wide failure due to a single node's downtime or malicious attacks. By establishing a distributed storage network, the system gains increased resilience and eliminates the susceptibility to a single point of failure. This approach not only disperses power within the system but also enhances data security, facilitating data sharing and improving overall system efficiency.

Privacy protection: in the proposed scheme, all EMR operations conducted by patients and doctors are contained within the system, eliminating external system node involvement and safeguarding patients' sensitive data from exposure. Additionally, the patient's sensitive data is encrypted using the system's SM4 symmetric encryption and SM2 signature algorithm. Adversaries cannot decrypt and access the EMR ciphertext without the corresponding symmetric key and owner's signature. Furthermore, the EMR is stored

```
Input: Role, userAddress
Output: Bool
  1: if !hasRole(Role, userAddress) then
       revert string abi.encodePacked(userAddress,uint(Role), 32)
  2: else
  3:   return false
  4: end if
```

Algorithm 6: Role access rights determination.

within the IPFS storage network, and the IPFS network's address value is securely uploaded to the blockchain using SM4 symmetric encryption. The entire system operates on a private chain, permitting blockchain access only to authorized users vetted by the system, ensuring the safety and security of patient-sensitive data within the system.

Data integrity: the system guarantees the tamper-resistant nature of EMR data and patient's private information through a dual backup mechanism. Initially, the system generates two data copies: one is uploaded to the internal IPFS network and stored on the institution's blockchain. Additionally, the patient's private data is encrypted and securely stored to prevent unauthorized access or data leakage, enhancing the security of sensitive patient details. Furthermore, when medical institutions or doctors access the data, the system enforces a dual verification process to ensure data authenticity and integrity, effectively thwarting any malicious tampering.

Resistance to double-spend attacks: in our proposed scheme, all EMR data transactions involving MedCoin tokens are diligently recorded on the system's blockchain, including details of the sender, recipient, and token quantity in each transaction. This robust tracking system guarantees the traceability of all MedCoin token transactions within the system, preventing users from engaging in double spending. The system can detect and block the reuse of tokens already expended in prior transactions, ensuring the security of MedCoin tokens.

Protection against DDoS attacks: DDoS attacks, which can disrupt system operations by inundating resources with a high volume of requests, are a significant concern. The proposed scheme in this paper tackles this problem by implementing a distributed system architecture. Distributing the workload across multiple nodes mitigates the risk of resource exhaustion in any one node, safeguarding the system against potential collapse due to resource depletion. This distributed approach effectively protects the system from DDoS attacks, ensuring uninterrupted operation.

Resistance to masquerade attacks: attackers discover system vulnerabilities by obtaining user credentials and user passwords and use masquerade attacks to gain unauthorized access control to the system. Since the privilege control of our scheme is based on RBAC, this system is resistant to masquerade attacks. Before connecting to the system network, all the entities in the system have to register in the RegisterUser() function and provide the necessary data, and then, the system authenticates them. In addition, the system encrypts the EMR, resulting in a ciphertext CT that cannot be decrypted without the user's authorization. Finally, the hash value of the EHR is stored on the blockchain and only users who satisfy the set of policy attributes can decrypt the EMR, which is inaccessible to attackers.

Resistance to man-in-the-middle attacks: in the data-sharing process, the attacker somehow obtains the hash value of the data and tries to access the data in the IPFS but cannot access the data in the IPFS because the attacker must also obtain the relevant signatures for signature verification. The attacker is unable to obtain the signatures stored on the chain normally through the access control policy.

5.3. Performance Analysis

5.3.1. Gas Fee Consumption. This section primarily focuses on analyzing the gas fee consumption for deploying the paper's scheme on the Ethereum network. Deploying a contract or executing transactions on Ethereum incurs transaction fees, which correspond to the cost of processing data on the blockchain. Table 4 presents the gas fee consumption for key functions during each major phase of the system and analyzes the cost of smart contract creation and execution for the main methods. As of December 2022, considering the current Ethereum conversion rate of 1 gas ≈0.000000001 eth, the primary source of gas fee consumption in the system stems from the initial deployment of smart contracts. On the other hand, the gas fee generated during the execution of smart contracts is comparatively lower for the system. Table 5 mainly shows a comparative comparison of the gas fee consumption of some of the functions with literature [20].

5.3.2. Trading Latency Analysis. Transaction latency analysis examines the time required to complete a specific function within the system, and this completion time is a crucial factor for evaluating system performance. Transaction latency encompasses the entire process, starting from initiating a transaction to its confirmation, which includes the time taken for transaction broadcasting and the execution time of the consensus algorithm [21].

$$T_{\mathrm{LT}} = (T_{\mathrm{CT}} \cdot T_{\mathrm{NT}}) - T_{\mathrm{ST}}, \tag{8}$$

where T_{LT}, T_{CT}, T_{NT}, and T_{ST} are the transaction delay, transaction confirmation time, network threshold, and transaction submission time, respectively.

Hence, the performance evaluation of the proposed system can be demonstrated by conducting a latency analysis of its major functions. In this section, we provide a brief

Table 4: Consumption of gas fee for the method in the scheme.

Function name	Gas used	Actual Tx cost
Create Contract	1419799	0.001419799
InitializeEMR	168010	0.00016801
AddEMR	133329	0.000133329
RequestEMR	142320	0.00014232
AddDoctor	44721	0.000044721
AddPatient	344709	0.000344709
AddUser	145590	0.00014559
RemoveUser	101819	0.000101819
VerifyRole	120910	0.00012091

Table 5: Comparison table of gas fee consumption of some functions in the scheme with literature [20].

Function name	Proposed model gas used	SVBE [20] gas used
AddPatient	344709	421901
AddFile	92858	93968

calculation of transaction latency for several key functions in the scheme.

Firstly, the processing time related to patients and medical records is illustrated in Figure 8. It can be observed that the transaction delay time for patient identity verification is the shortest. In contrast, the system delay time for filling out medical record information and searching medical records is longer. It can be influenced by the size of the electronic medical record file. Specifically, larger electronic medical record files result in longer system delay times for these two tasks. Compared to alternative solutions, this solution demonstrates faster performance in adding and verifying patients, as well as adding EMRs.

Secondly, the system processing time for IPFS file upload and validation in the statistical system is depicted in Figure 9. We conducted tests with different file sizes, including 100 MB, 300 MB, 500 MB, and 1000 MB. As the file size increases, the system processing time for IPFS file upload and validation also increases. The proposed system exhibits certain advantages over other systems regarding transaction latency.

5.3.3. Computational Overhead. In this section, we assess the computational overhead of the proposed model, with a specific focus on the decryption and encryption computational overhead on the user side. To alleviate the computational burden on users, the system employs a strategy that primarily performs decryption and encryption computations, and the results are transmitted to the user side. Compared with existing schemes such as [11], [38], and [39], the proposed scheme in this paper demonstrates certain advantages in terms of decryption and encryption time. Furthermore, the user side's decryption and encryption computation overhead does not significantly increase, as depicted in Figures 10 and 11. It is worth noting that the system bears the main burden of decryption and encryption computation, effectively reducing the computational pressure on the user side.

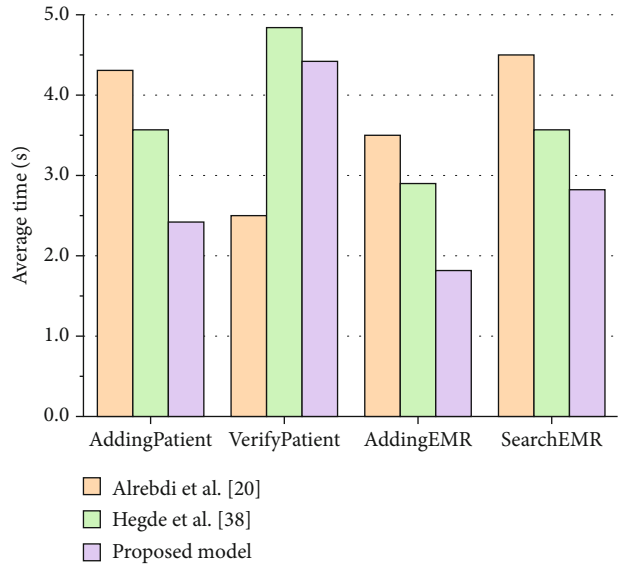

Figure 8: Transaction latency of functions.

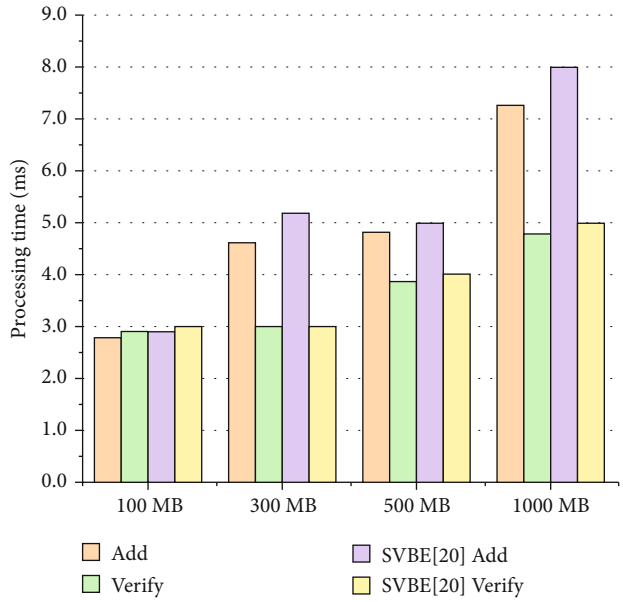

Figure 9: Transaction latency of adding and verifying files.

Next, we examine the computational overhead of running the signature algorithm. For this experiment, we utilize data sizes ranging from 100 MB to 1000 MB, and the results are presented in Figure 12. The computational overhead of the signature algorithm employed in this paper exhibits a linear increase as the data size of the electronic medical record (EMR) grows. Notably, our proposed scheme demonstrates an advantage over the approaches described in literature [40], literature [41], literature [42], and literature [9] in terms of the computational overhead of the signature algorithm.

The last is an analysis of the algorithmic overhead of the whole scheme compared to other schemes, and the symbols used in this section are defined in Table 6.

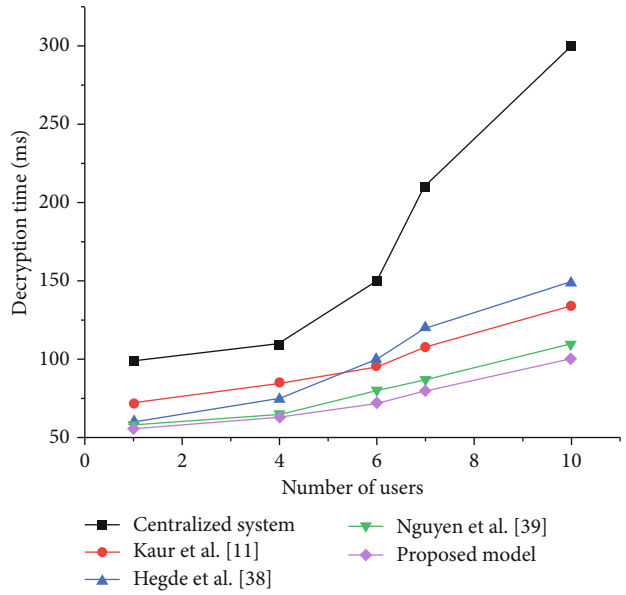

Figure 10: Decryption time on the user side.

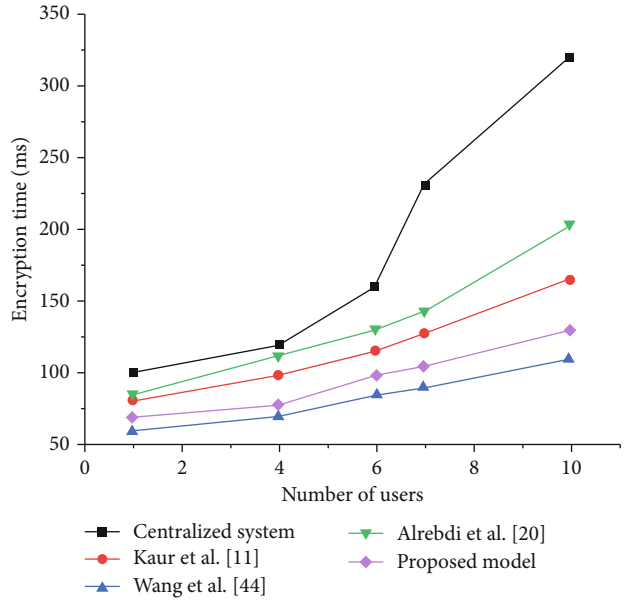

Figure 11: Encryption time on the user side.

As shown in Table 7, the overall computation overhead in encryption and decryption operations is much lower than that of scheme [7], which is $(4n+5)\cdot E_G + (n+2)\cdot \text{Hash}_G + (4n+2)\cdot \text{Mult}_G + (2n+1)\cdot P$. In the encryption phase, the computation overhead of this paper is $(n+1)\cdot E_G + n\cdot \text{Hash}_G + (n+1)\cdot \text{Mult}_G + n\cdot P$, while the computational overhead of scheme [43] in the encryption phase is $P + (n+1)\cdot E_G + (n+4)\cdot \text{Hash}_G + (2n+5)\cdot \text{Mult}_G + E_P$, and it can be found that the computational overhead of this paper in the encryption phase is lower than that of scheme [42]. From the data provided in Table 7, it can be found that the overall computational overhead of this paper is also lower than that of schemes [20, 21, 44] and

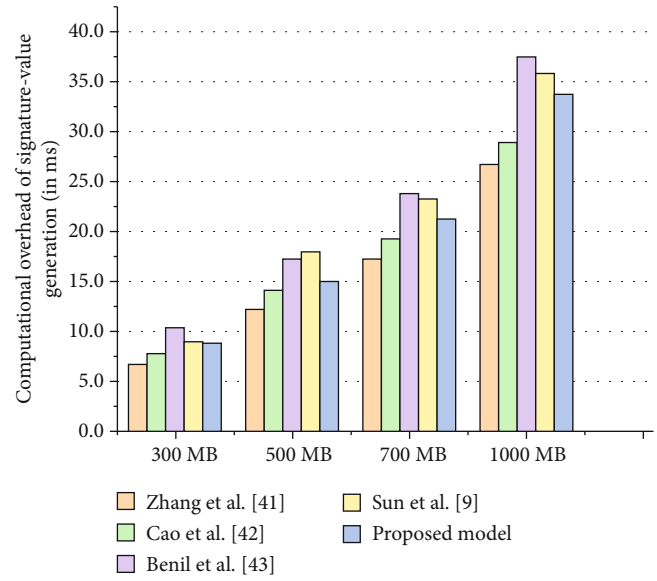

Figure 12: Computational overhead of signature-value generation.

Table 6: Algorithm overhead symbol definition table.

Notation	Description
E_G	The exponential operations in group G
Hash_G	Hash function in group G
Mult_G	Point multiplication in group G
P	The pairing operations
E_P	The exponential operations in ring

[45]. Therefore, the scheme designed in this paper has an advantage in overall computational overhead compared to other schemes.

5.3.4. Transaction Throughput. Transaction throughput is the number of successful transactions completed on the blockchain in a specific period. When performing the statistics, invalid transactions among them should be excluded from the total number of transactions to obtain successful transactions. The transaction throughput can be expressed as the following equation [46]:

$$\text{Transaction throughput} = \frac{\text{successful} - \text{transactions}}{\text{time (sec)}}. \quad (9)$$

In blockchain, the transaction delivery rate is the number of transactions sent to the blockchain network per unit of time. A blockchain network is a peer-to-peer network of nodes distributed around the globe, where each node can receive, process, and broadcast transactions. When a user initiates a transaction on the blockchain network, that transaction is sent to a nearby node and progressively broadcast to other nodes throughout the network. Therefore, the transaction delivery rate is one of the important measures of the

Table 7: Comparison of computational overheads.

Schemes	Overall computation over
[7]	$(4n+5)\cdot E_G + (n+2)\cdot \text{Hash}_G + (4n+2)\cdot \text{Mult}_G + (2n+1)\cdot P$
[20]	$(3n+5)\cdot E_G + (3n+2)\cdot \text{Hash}_G + (n+6)\cdot \text{Mult}_G + (2n+7)\cdot P$
[21]	$(n+2)\cdot E_G + (3n+2)\cdot \text{Hash}_G + (4n+4)\cdot \text{Mult}_G + (2n+4)\cdot P$
[42]	$(2n+1)\cdot E_G + 7n\cdot \text{Hash}_G + (4n+5)\cdot \text{Mult}_G + 4n\cdot P$
[44]	$P + (2n+2)\cdot E_G + (2n+5)\cdot \text{Hash}_G + (2n+5)\cdot \text{Mult}_G + 2\cdot E_P$
[45]	$(3n+3)\cdot E_G + (3n+4)\cdot \text{Hash}_G + 4n\cdot \text{Mult}_G + (3n+1)\cdot P$
Ours	$(2n+1)\cdot E_G + (n+1)\cdot \text{Hash}_G + (3n+2)\cdot \text{Mult}_G + 2n\cdot P$

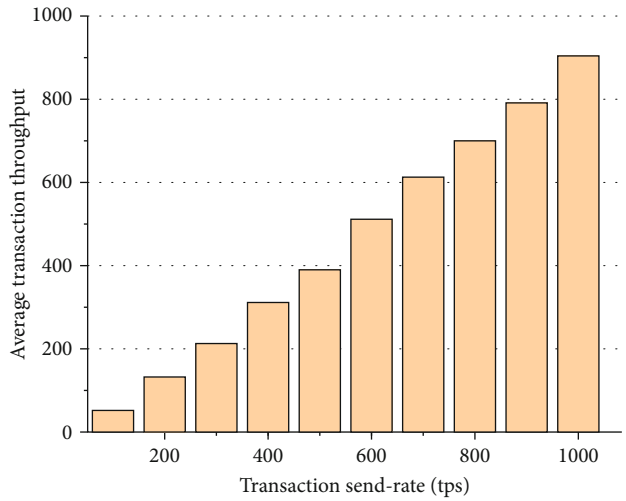

Figure 13: Average transaction throughput of the proposed model.

transaction traffic that a blockchain network can handle and can be expressed as the following equation:

$$\text{Transaction\_sendRate} = \frac{\text{Transactions\_number}}{\text{Block\_time}} \cdot \text{Average\_Transaction\_Size}, \tag{10}$$

where Transactions number refers to the total number of transactions submitted to the blockchain network in a certain time, Block time is the average time for the blockchain network to generate new blocks, and Average Transaction Size refers to the average number of bytes of transaction data.

In this paper, the size of the average transaction throughput of the model is calculated by setting a series of transaction sending rates. The average transaction throughput of the model in this paper increases as the transaction sending rate increases, and the average transaction throughput increases, as shown in Figure 13.

5.4. Functional Analysis. In Table 8, the characteristics of some of the schemes are compared, and the scheme proposed in this paper has certain advantages compared to other schemes.

Table 8: Comparison of key functions of blockchain-based electronic medical data systems.

Schemes	Privacy protection	Access control	Blockchain	IPFS	Incentives
[16]	✓	✓	✓	✓	x
[17]	✓	✓	✓	x	✓
[24]	x	x	x	x	x
[25]	x	✓	x	x	x
[26]	✓	x	✓	x	x
[29]	✓	✓	✓	x	x
[30]	✓	✓	✓	x	x
[43]	✓	✓	✓	x	x
[20]	✓	✓	✓	✓	x
[21]	✓	✓	✓	✓	x
[22]	✓	✓	✓	x	x
Ours	✓	✓	✓	✓	✓

6. Conclusion and Future Work

This paper introduces a novel sharing scheme that leverages blockchain and IPFS technologies to facilitate the secure and reliable sharing of electronic medical data among various entities. The scheme incorporates an incentive mechanism by evaluating the potential value of electronic medical records, encouraging patients to share their records, and enhancing the efficiency of data circulation within the electronic medical data system. Additionally, the proposal utilizes blockchain technology to ensure the integrity of electronic medical records, making them tamper-proof and enabling traceability of patient data. Future research will focus on enhancing the system's access control to achieve a more granular level of control and security.

In reality, Ethereum-based electronic medical record solutions have inherent shortcomings, and the relatively low transaction speed of Ethernet may lead to performance bottlenecks in this system, resulting in the inability to cope with higher system throughput and leading to the exhaustion of system resources. Therefore, we will further explore an extensible coalition chain blockchain architecture that also supports the deletion of blocks. This extensible federated chain blockchain architecture can support efficient storage management on the blockchain with invariance and high

extensibility. In terms of access control, this scheme relies on smart contracts to realize access control, but the finesse of access control is not enough, and the next step will be to investigate the finesse of access control.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. C. A. Pilares, S. Azam, S. Akbulut, M. Jonkman, and B. Shanmugam, "Addressing the challenges of electronic health records using blockchain and ipfs," *Sensors*, vol. 22, no. 11, p. 4032, 2022.

[2] A. N. Khan, M. M. Kiah, M. Ali, S. A. Madani, A. U. R. Khan, and S. Shamshirband, "BSS: block-based sharing scheme for secure data storage services in mobile cloud environment," *The Journal of Supercomputing*, vol. 70, no. 2, pp. 946–976, 2014.

[3] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *Journal of Medical Systems*, vol. 43, no. 1, pp. 1–9, 2019.

[4] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *Journal of Information Security and Applications*, vol. 50, article 102407, 2020.

[5] T. F. Stafford and H. Treiblmaier, "Characteristics of a blockchain ecosystem for secure and sharable electronic medical records," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1340–1362, 2020.

[6] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," in *2016 2nd International Conference on Open and Big Data (OBD)*, pp. 25–30, Vienna, Austria, 2016.

[7] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K.-C. Li, "A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3582–3592, 2019.

[8] X. Zhang and S. Poslad, "Blockchain Support for Flexible Queries with Granular Access Control to Electronic Medical Records (EMR)," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, 2018.

[9] J. Sun, X. Yao, S. 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.

[10] J. Sun, L. Ren, S. Wang, and X. Yao, "A blockchain-based framework for electronic medical records sharing with fine-grained access control," *Plos One*, vol. 15, no. 10, article e0239946, 2020.

[11] J. Kaur, R. Rani, and N. Kalra, "Attribute-based access control scheme for secure storage and sharing of EHRs using blockchain and IPFS," *Cluster Computing*, pp. 1–15, 2023.

[12] Y. Cheng, B. Gong, Z. Jia, Y. Yang, Y. He, and X. Zhang, "Efficient and secure cross-domain sharing of blockchain electronic medical records based on edge computing," *Security and Communication Networks*, vol. 2021, Article ID 7310771, 10 pages, 2021.

[13] R. Thilagavathy, P. Renjith, R. Lalitha, M. Y. B. Murthy, Y. Sucharitha, and S. L. Narayanan, "A novel framework paradigm for emr management cloud system authentication using blockchain security network," *Soft Computing*, pp. 1–9, 2023.

[14] S. Fugkeaw, L. Wirz, and L. Hak, "Secure and Lightweight Blockchain-Enabled Access Control for Fog-Assisted Iot Cloud Based Electronic Medical Records Sharing," *IEEE Access*, vol. 11, pp. 62998–63012, 2023.

[15] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin*, vol. 4, no. 2, p. 15, 2008, https://bitcoin.org/bitcoin.pdf.

[16] C. Li, M. Dong, J. Li, G. Xu, X. Chen, and K. Ota, "Healthchain: secure EMRs management and trading in distributed healthcare service system," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7192–7202, 2020.

[17] C. A. Holt and A. E. Roth, "The Nash equilibrium: a perspective," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 3999–4002, 2004.

[18] J. Liu, Y. Fan, R. Sun, L. Liu, C. Wu, and S. Mumtaz, "Blockchain-Aided Privacy-Preserving Medical Data Sharing Scheme for e-Healthcare System," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21377–21388, 2023.

[19] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 152–167, 2022.

[20] N. Alrebdi, A. Alabdulatif, C. Iwendi, and Z. Lian, "SVBE: searchable and verifiable blockchain-based electronic medical records system," *Scientific Reports*, vol. 12, no. 1, p. 266, 2022.

[21] D. Ramesh, R. Mishra, P. K. Atrey, D. R. Edla, S. Misra, and L. Qi, "Blockchain based efficient tamper-proof EHR storage for decentralized cloud-assisted storage," *Alexandria Engineering Journal*, vol. 68, pp. 205–226, 2023.

[22] Z. H. Mohammed, K. Chankaew, R. R. Vallabhuni, V. R. Sonawane, S. Ambala, and S. Markkandan, "Blockchain-enabled bioacoustics signal authentication for cloud-based electronic medical records," *Measurement: Sensors*, vol. 26, article 100706, 2023.

[23] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[24] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.

[25] W. Wang, H. Huang, Z. Yin, T. R. Gadekallu, M. Alazab, and C. Su, "Smart contract token-based privacy-preserving access control system for industrial Internet of Things," *Digital Communications and Networks*, vol. 9, no. 2, pp. 337–346, 2023.

[26] D. D. F. Maesa, A. Lisi, P. Mori, L. Ricci, and G. Boschi, "Self sovereign and blockchain based access control: supporting attributes privacy with zero knowledge," *Journal of Network and Computer Applications*, vol. 212, article 103577, 2023.

[27] N. Szabo, "Smart contracts: building blocks for digital markets," *EXTROPY: The Journal of Transhumanist Thought, (16)*, vol. 18, no. 2, p. 28, 1996.

[28] P. Sharma, R. Jindal, and M. D. Borah, "A review of smart contract-based platforms, applications, and challenges," *Cluster Computing*, vol. 26, no. 1, pp. 395–421, 2023.

[29] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[30] A. S. Rajasekaran, M. Azees, and F. Al-Turjman, "A comprehensive survey on blockchain technology," *Sustainable Energy Technologies and Assessments*, vol. 52, article 102039, 2022.

[31] V. Buterin, "A next-generation smart contract and decentralized application platform," *White Paper*, vol. 3, no. 37, 2014.

[32] J. Benet, "Ipfs-content addressed, versioned, p2p file system," 2014, http://arxiv.org/abs/1407.3561.

[33] L. Bai, Y. Zhang, and G. Yang, "SM2 cryptographic algorithm based on discrete logarithm problem and prospect," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 1294–1297, Yichang, China, 2012.

[34] Z. Wang, H. Dong, Y. Chi, J. Zhang, T. Yang, and Q. Liu, "Research and Implementation of Hybrid Encryption System Based on SM2 and SM4 Algorithm," in *Proceedings of the 9th International Conference on Computer Engineering and Networks*, Q. Liu, X. Liu, L. Li, H. Zhou, and H. H. Zhao, Eds., vol. 1143 of Advances in Intelligent Systems and Computing, Springer, Singapore, 2021.

[35] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (rbac): features and motivations," in *Proceedings of 11th annual computer security application conference*, pp. 241–248, New Orleans, LA, 1995.

[36] E. Coyne and T. R. Weil, "ABAC and RBAC: scalable, flexible, and auditable access management," *IT Professional*, vol. 15, no. 3, pp. 14–16, 2013.

[37] P. Esmaeilzadeh and T. Mirzaei, "Role of incentives in the use of blockchain-based platforms for sharing sensitive health data: experimental study," *Journal of Medical Internet Research*, vol. 25, article e41805, 2023.

[38] M. Hegde, R. R. Rao, and B. Nikhil, "Ddmia: distributed dynamic mutual identity authentication for referrals in blockchain-based health care networks," *IEEE Access*, vol. 10, pp. 78557–78575, 2022.

[39] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure ehrs sharing of mobile cloud based E-health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019.

[40] X. Zhang, J. Zhao, C. Xu, H. Li, H. Wang, and Y. Zhang, "CIPPPA: conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors," *IEEE transactions on cloud Computing*, vol. 9, no. 4, pp. 1362–1375, 2019.

[41] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, "Cloudassisted secure ehealth systems for tamper-proofing EHR via blockchain," *Information Sciences*, vol. 485, pp. 427–440, 2019.

[42] T. Benil and J. Jasper, "Cloud based security on outsourcing using blockchain in e-health systems," *Computer Networks*, vol. 178, article 107344, 2020.

[43] L. Hong, K. Zhang, J. Gong, and H. Qian, "A practical and efficient blockchain-assisted attribute-based encryption scheme for access control and data sharing," *Security and Communication Networks*, vol. 2022, Article ID 4978802, 14 pages, 2022.

[44] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020.

[45] H. Wang and Y. Song, "Secure cloud-based ehr system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 42, no. 8, p. 152, 2018.

[46] H. M. Hussien, S. M. Yasin, N. I. Udzir, and M. I. H. Ninggal, "Blockchain-based access control scheme for secure shared personal health records over decentralised storage," *Sensors*, vol. 21, no. 7, p. 2462, 2021.