International Journal of Advanced Science and Computer Applications (2024); 4(2)
DOI: 10.47679/ijasca.v4i2.71
https://ijasca.org/index.php/ijasca

RESEARCH ARTICLE

A Secure Storage For Medical Information Scheme Using Blockchain

Adoni Kadjo Mathias; Yuan Xu

Published online: 15 May 2024

Abstract

Nowadays, many companies, organizations, hospitals and individuals have adopted centralized data storage systems to store and share data. However, these systems create a single point of failure and involve a centralized entity or third party, which can cause concern for users. Decentralized storage systems are therefore needed to overcome the drawbacks of the traditional approach. However, in the face of centralization issues, this paper proposes a combination of Hyperledger Fabric, InterPlanetary File System (IPFS), Attribute-Based Access Control (ABAC), and proxy re-encryption to enhance the security and transparency features of decentralized storage systems. Thus, the proposed scheme provides a secure decentralized system storage of medical information using a consortium blockchain.

Keyword: Blockchain, IPFS, ABAC, Proxy re-encryption , hyperledger fabric , storage medical information

INTRODUCTION

With the development of technology, various emerging technologies are merging with the healthcare sector, making the process of building healthcare information technology increasingly challenging [1]. The World Health Organization defines medical information as the most innovative and shared asset. Nowadays, the number of medical institutions around the world shows an index stage growth, and the medical data generated by medical institutions also shows an explosive growth. Due to the deepening of the degree of information in hospital information, the information system within the hospital gradually expands from a single HIS billing system to a system with electronic medical records. system into an electronic medical record system. The medical data is accompanied by the registration, diagnosis and hospitalization, medical data is gradually complex and stereochemical, and the importance of privacy and security is greatly increased [2]. At present, the combination of traditional paper medical records and centralized medical data management systems is still the primary method used by healthcare organizations to track patient medical information (Figure 1). However, there are significant privacy concerns with this type of medical system [3]. As a result, the shift from centralized medical data management to distributed medical data sharing is a trend that the entire society is bound to follow[4]. Data islands are created when medical and health institutions store and manage medical health data apart from one another due to their isolation from one another. This not only makes it difficult to keep long-term records of patients and the progression of their diseases, but it also results in a waste of medical equipment and a significant amount of duplicate medical data. Data sharing across medical institutions is a trend that is unavoidable in order to maximize the value of medical health data, satisfy the fundamental requirements of medical information creation, and offer patients more reasonable and compassionate services [5]. Furthermore, as a result of the medical industry's widespread use of cutting-edge Internet technology, medical data transmission pathways and techniques have expanded in variety and steadily moved from the internal transmission of The transmission of patient data between hospitals, medical institutions,insurance companies, and other organizations, as well as between patients and medical institutions, poses a significant challenge to patient data privacy [6]. The aforementioned factors contribute to the vast volume, intricate structure, and quick expansion of medical data, making the selection of the most effective storage strategy challenging

Dalian University of Technology, China

*) corresponding author

Email: donikadjo@hotmail.com

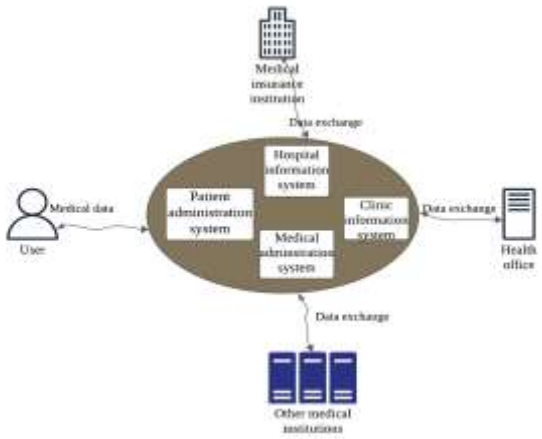

Fig 1. Medical information exchange

Fortunately, in recent times, the emergence of blockchain technology has introduced novel remedies for the safeguarding of medical data. Additionally, blockchain is a distributed database that encompasses characteristics such as decentralization, security, and transparency. In its capacity as a decentralized database, blockchain also presents a dependable resolution to the challenges associated with the sharing, efficacy, and security in the management of medical information. However, the application of blockchain technology in the medical field has been unsatisfactory. In this article, we store medical data inconsortium blockchain by deploying smart contracts in Hyperledger fabric to guarantee the confidentiality and security of medical data. At the same time, the ABAC model is being introduced for access control to ensure that users can access it securely and efficiently. In addition, due to the enormity and complexity of medical information, we also combine proxy re-encryption to encrypt medical information and the interstellar file system to alleviate blockchain storage pressure and improve user access efficiency to alleviate blockchain storage pressure. Specifically, the main contributions of this study are follow :

1. This study employs blockchain technology for the management of medical information, achieving decentralized management and secure storage through the utilization of distributed consensus and authentication mechanisms.
2. A distributed storage system (IPFS) has been adopted. Medical data files are encrypted and stored in IPFS. It is therefore not necessary to place the data itself on the blockchain, which not only saves bandwidth on the blockchain network, but also bridges the gap of limited file storage in the existing blockchain system, improving throughput and scalability.
3. Proxy re-encryption technology combined with ABAC technology not only guarantees the security of data storage in IPFS, but also ensures fine-grained access control to medical information.
4. We create a backup architecture based on ABAC that enables dynamic permission management and fine-grained access control.

Related Work

At present, blockchain technology has been widely applied in finance, Internet of Thing (IoT) and other fields by taking advantage of its decentralization, non-tampering and distributed storage features. With the gradual improvement of blockchain technology, its application in the medical field has also made rapid progress. People believe that the medical and health field is the second largest field of research outside of the financial field. Shrier and Chang et al. used MIT's OPAL/Enigma encryption platform combined with blockchain technology to create a secure environment for storing and analyzing medical data [6]. Based on blockchain private chain, Kuo T et al. build a cross agency medical health prediction model [7]. The literature [8,9] focuses on current problems of serious illnesses fragmentation of medical data, low sharing efficiency, insecure transmission process, lack of data integrity verification and insufficient protection of confidential information, access and sharing of medical information of data are carried out via the Ethereum platform with smart contract. The MedRec framework mentioned in the literature [10,11] combines a smart contract with access control for automatic authorization management, which realizes the integration and authorization management of medical data distributed in different organizations. Since the MedRec framework uses the PoW consensus mechanism to maintain blockchain consistency, the required computational load is too large. Therefore, the MDSN[11] framework innovates the consensus mechanism by using the DPoS consensus mechanism to reduce resource consumption, and the framework adopts the method of asymmetric encryption and proxy re-encryption to control access to medical data, which improves the efficiency of data sharing while protecting privacy, but it also has the deficit of limited data storage capacity. However, the systems [13–15] are all implemented on Ethereum, whereas Rajput et al. [16] pointed out that the Ethereum system suffers from the weaknesses of inefcient transactions and higher energy consumption compared to Hyperledger Fabric. Therefore , this article designs a blockchain consortium model for storing and sharing medical data. we survey blockchain-based secure storage in section II-A and blockchain-based secure sharing in section II-B

Bloccckchain-based secure storage of medical data

The impact of blockchain technology on the healthcare field is significant due to its decentralized, tamper-proof, and transparent nature. In their work, Azaria et al. [28] propose a decentralized MedRec system based on blockchain technology to handle Electronic Health Records (EHR). MedRec utilizes a modular design where system participants have administrative privileges, authorization, and data sharing capabilities. Another approach, Medblock [29], employs a hybrid blockchain architecture to safeguard Electronic Medical Records (EMR). This architecture consists of endorsement nodes, sorting nodes, and submission nodes. The consensus algorithm used is a variant of the part consensus algorithm. In their research,

Conceic¸˜ao et al. [30] present a generic architecture that leverages blockchain technology to store patient EHR data. Yang and Li [31] propose an EHR architecture that is based on blockchain and effectively prevents tampering and abuse of EHR by tracking all events in the blockchain. Kushch et al. [32] introduce a specialized data structure called the blockchain tree for storing electronic medical data on the blockchain. This structure includes a subchain, recorded patient identity, and additional critical information (e.g., diagnostic records), with blocks on the main chain serving as initial blocks of the sub-chain.

Bloccckchain-based secure sharing of medical data

Moreover, to its application in secure storage, the blockchain is also widely utilized in the realm of security sharing. The medical field has placed significant emphasis on the utilization and exploration of blockchain technology in the management of medical records, prompting numerous research institutions worldwide to actively engage in this area of study. Xia et al. [33] introduced a blockchain-based system known as "men shared," which effectively minimizes the risk of compromising data privacy and offers a viable solution for medical data sharing amongst custodians operating within an environment lacking trust. Zhang et al. [34] proposed a medical data sharing scheme that leverages the hospital's private blockchain to store patients' health data, while employing the consortium blockchain to securely store the corresponding security index. Furthermore, Zhang et al. [35] combined artificial intelligence technology with blockchain technology to present a secure and transparent platform for medical data sharing. This platform capitalizes on the transparency of the zone chain to enable data tracking, thus ensuring the integrity and immutability of the shared information. To address privacy concerns within the medical field, Liu et al. [36] developed a data sharing scheme that utilizes both blockchain technology and cloud storage technology. This scheme involves storing the original medical data in the cloud, indexing the data within the blockchain, and leveraging the tamper-proof nature of the blockchain to prevent malicious modifications to the data. Lastly, Qiao et al. [37] proposed a scheme that facilitates dynamic communication between healthcare alliance chains

Preliminaries

In this section, we present the architecture of secure medical information storage based on blockchain and access control with proxy re-encryption. section III-A presents the system structure, section III-B presents the workflow, section III-C and section III-D explains the design of the smart contract.

Knowledge of blockchain

Since 2008 Bitcoin was proposed by Satoshi Nakamoto, blockchain has been widely noticed. Bitcoin is still one of the representatives of cryptocurrency . The chain structure, Merkle tree and hash algorithm together guarantee the blockchain's tamper-evident nature. All nodes maintain the same ledger together, ensuring decentralization. Because of asymmetric encryption and authorization technology, the transaction information stored on the blockchain is public, and the account identification information is highly encrypted. Access is only possible with the authorization of the data owner, which ensures data security and personal privacy.

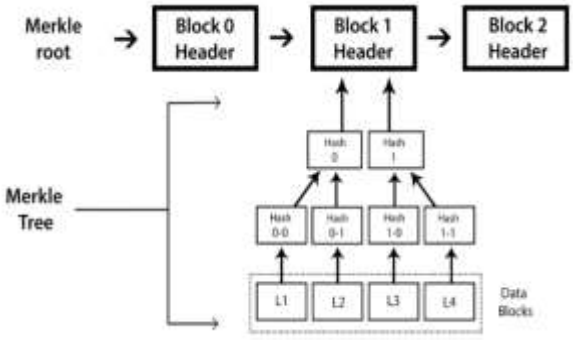

Fig 2. Blockchain structure

Importance of blockchain technology in Healthcare innformation storage

Blockchain technology facilitates the establishment of decentralized mechanisms for sharing data and applications. In the realm of medical information management, a traditionally unilateral approach is employed. However, the centralized nature of this mode of management poses challenges in achieving genuine information sharing. The introduction of blockchain technology brings forth the concept of distributed ledgers. Under this technology, the input of file information is collectively maintained and supervised by multiple parties, ensuring the openness and transparency of data information. Additionally, the rules governing blockchain transactions are determined by the joint supervision of various information data by multiple departments. Consequently, this innovative approach addresses the issues of low work efficiency and disorder prevalent in traditional medical information management. Furthermore, blockchain has the capability to establish a trusted deposit system. The management of medical archives information primarily revolves around four fundamental processes: addition, deletion, modification, and query. However, in the context of blockchain, the processes of deletion and modification in archives information management are discarded, reducing the complexity of the information processing workflow. The technical design of blockchain guarantees the irreparability and security of data information. Moreover, each block of information within the blockchain meticulously records the creation time and the hash value of the preceding block. This network structure, which is characterized by its association with time, facilitates the customary scrutiny, monitoring, and verifiability, and enhances the rate of utilization of medical data. Ultimately, the blockchain can consolidate the regulations governing data exchange and the allocation of benefits. The convergence of smart contracts and blockchain technology can

optimize the automation of sharing archival information. Once the implementation of a smart contract is initiated, it becomes impervious to cessation and remains unaffected by external operations. Hospitals can exploit this characteristic to firmly establish rules regarding the distribution of interests. In the context of medical data sharing, smart contracts can transform the conduct of participants engaged in data sharing, promoting active involvement, enhancing the efficiency and celerity of information dissemination, and effectively maximizing the worth of medical information. In mandatory information sharing, the manner in which the secret box operates in traditional information sharing is restricted, ensuring the quality of medical data information

Hyperledger fabric

Nowaday, cryptocurrencies, exemplified by Bitcoin, have attained significant triumph, effectively capturing the global attention towards blockchain technology. Nevertheless, these public chains encounter several issues, such as limited transaction capacity, extended transaction durations, squandered resources, and data consistency concerns. In order to rectify these predicaments, the Linux Foundation initiated the Hyperledger project in 2015, which stands as one of the largest blockchain projects worldwide and frequently serves as a platform for enterprise blockchain development. Hyperledger Fabric is constructed with a modular framework encompassing members, blockchain, transactions, and smart contracts, as illustrated in Figure 2. The member management module caters to the requirements of enterprise-level blockchains, ensuring security and privacy. It reinforces the user's joining permissions and necessitates certification through the PKI public key infrastructure for any party involved in the transaction. The blockchain module employs the P2P protocol to govern distributed ledgers and can configure distinct consensus protocols to accommodate various demands. It categorizes transaction history within the chain and presents the latest state through the World State mechanism, as depicted in Figure 3. The transaction module governs data within the transaction process via deployment transactions and invocation transactions. Deployment transactions install Chaincode on all peer nodes upon

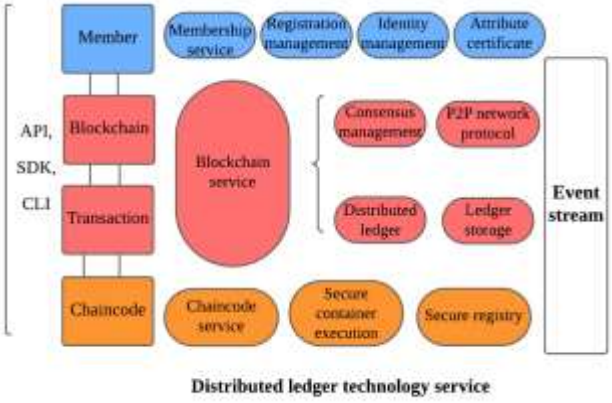

Fig 3. Architecture diagram of Hyperledger fabric

successful execution of the transaction, while invocation transactions are carried out by invoking specified functions in the Chaincode using the Fabric Software Development Kit's provided SDK. Smart contracts record the mutually agreed business logic among Fabric's federated chain members and can be scripted in widely used programming languages like Go and Java, thereby surmounting the limitations of traditional blockchains that are restricted to domain-specific languages.

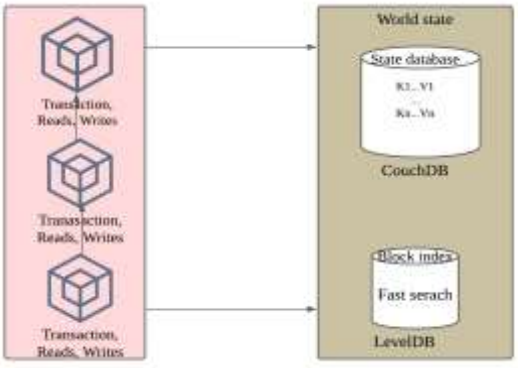

Fig 4. Architecture diagram of world state

Attribut-based access control model

Attribute-based access control involves a thorough consideration of user, resource, operation, and contextual access control policies. It focuses on determining whether access should be granted to a requester based on the correct attribute configuration. This approach does not require specifying the relationship between the data requester and the private data. Instead, it assesses whether the requester's attribute dictates access control permissions for the private data. The use of

attributes enhances the stability of the system operation. By utilizing attributes to define access control policies, one can segregate attribute management from the access decision phase. This separation allows for the adjustment and refinement of policies based on specific needs, facilitating updates, modifications, and the fine-tuning of access control granularity. Attributes serve as the foundation of the policy framework and can be categorized as a quadruplet A ∈ {S, O, P, E}. Each field in this quadruplet holds significance: A denotes attributes represented as key-value pairs; S pertains to subject attributes encompassing identity, role, position, and credentials; O refers to object attributes including identity, location, department, and data structure; E covers environmental attributes like time, system status, and security level; P relates to operation attributes primarily describing the subject's interactions with object types such as write, modify, and delete. The model's structure is illustrated in Fig. 4. An attribute-based access control request (ABACR) is defined as ABACR = {AS ∧ AO ∧ AP ∧ AE}, where AS represents subject attributes, AO denotes object attributes, AP signifies operation attributes, and AE indicates environmental attributes. Moreover, a set of rules denoted as R can also be defined as a quadruplet: R (A(Si), A(Oi), A(Ei), A(Pi)) → {Allow, Deny}. This formula signifies that the subject, with authorizing attribute Si, is attempting an access action Pi on object Oi within a contextual environment featuring attribute value Ei, leading to either an allow or deny outcome.

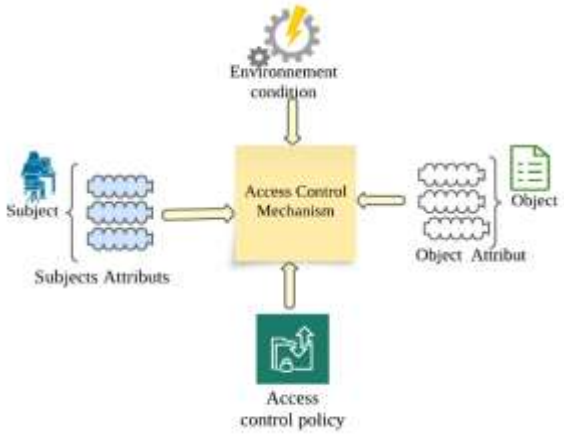

Fig 5. ABAC model

Proxy re-encryption model

Specifically, this technology allows user A to encrypt and upload ciphertext using a public key, and then transform the ciphertext into another format. Consequently, user B can decrypt the new ciphertext using his private key, while maintaining the confidentiality of the corresponding plaintext throughout the conversion process. In addition, proxy re-encryption provides a means to encrypt and decrypt data without directly exposing the plaintext to the data owner Fig 5. By using this technology, users have the ability to convert ciphertext into different formats, allowing other users to decrypt and obtain the plaintext using their private key. This method increases flexibility and security because the ciphertext can be decrypted by multiple users without revealing the contents of the plaintext.

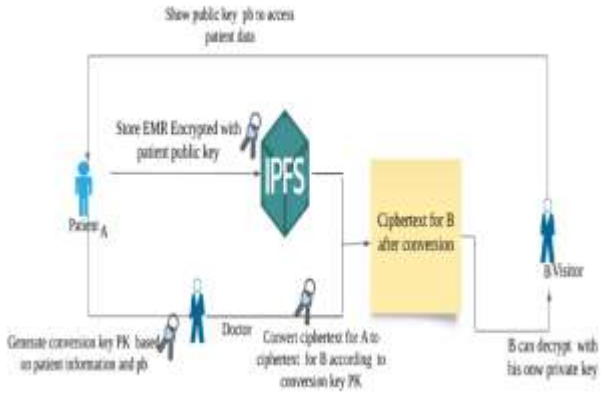

Fig 6. EMR Proxy re-encryption model

IPFS

Designed by Juan Benet and implemented by Protocol Labs in collaboration with the open-source community since 2014, IPFS (InterPlanetary File System) serves as a network transfer protocol aimed at establishing enduring and distributed storage and sharing of files. IPFS integrates elements from various existing technologies such as DHT, BitTorrent, Git, and SFS to fulfill its fundamental purpose of locally storing data and interconnecting nodes for data transmission. Initially conceived to construct a more robust resource network in comparison to the prevalent HTTP protocol, IPFS offers several advantages over HTTP, including rapid download speeds, global storage, security, and data perpetuation. Essentially, IPFS operates as a content-addressable, versioned, peer-to-peer hypermedia distributed storage and transport protocol, boasting distinctive features. Firstly, IPFS is content addressable, focusing solely on the file's content to produce a unique hash based on the

content, which is then utilized for retrieval and checked beforehand to determine if it already exists. This approach enables direct reading of stored files from other nodes without redundant storage, thus conserving space figure 6. Secondly, IPFS allows for the partitioning of large files without concern for their storage location or name, facilitating the simultaneous downloading of multiple partitions. Moreover, IPFS adopts a decentralized and distributed network structure, ideal for addressing storage limitations within blockchain by accommodating vast amounts of hypermedia data. Furthermore, IPFS incorporates encrypted storage by appending a unique cryptographic hash to digital data, ensuring the integrity of stored files as the hash remains unalterable and corresponds uniquely to the file. Within the IPFS network, considerations regarding server locations, file names, and paths are unnecessary. Each file stored in an IPFS node is assigned a distinct hash value derived from its contents, facilitating file retrieval based on the hash table. The amalgamation of IPFS with blockchain presents a promising solution to the storage challenges faced by blockchain technology.

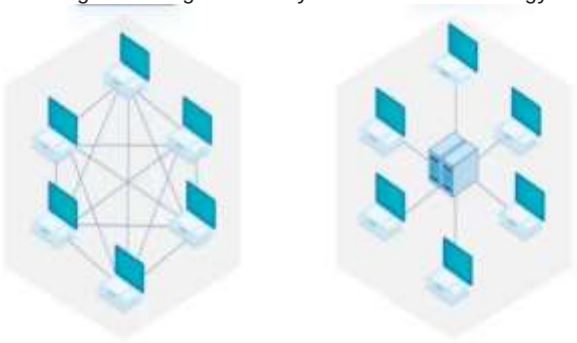

Fig 7. Peer to Peer model IPFS and Client server model HTTP

Methods

This section introduces the architecture of blockchain-based medical information security storage schemes and access controls. Section IV-A presents the system model, Section IV-B presents the workflow and Section IV-C describes the system's smart contract design.

System model

the architecture of the system consist of a user, attribut-based access control ,proxy re-encryption, IPFS and consortium blockchain show in figure 8.

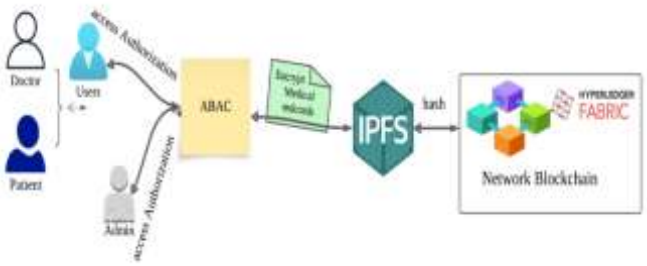

Fig 8. system model

Users can be classified into two categories: regular users, encompassing doctors and patients, both of whom are able to partake in the endorsement of the solution, thereby gaining access to medical information. Administrator users have the responsibility of overseeing the blockchain system and have the capability to generate or modify smart contracts. The amalgamation of the ABAC model with this framework constitutes a model for controlling access to medical information. The detailed outline of the model is presented below. (1) P = {AS, AO, AP, AE}

(2) AS = {userID, role, dep}

(3) AO = {recordId}

(4) AP = (1, allow

0, deny)

AE = {createTime, endTime}

Policy(P): It signifies the access control policy based on attributes that includes four components in the set, namely AS, AO, AP, and AE. Attribute of Subject(AS): This encompasses three key types of attributes, specifically user ID (identifying the unique identity of the user), user role (doctor and patient), and user department (specific department). Attribute of Object(AO): It involves the identification of the medical record ID (to ascertain the uniqueness of the record). Attribute of Permission(AP): This attribute indicates whether a user is authorized to access the medical record, with 1 indicating permission and 0 indicating denial. Attribute of Environment(AE): This component outlines the environmental prerequisites for the access control policy, primarily encompassing the creation time (when the policy was established) and the end time (when the policy expires). If the current time of a policy surpasses the end time, it implies that the policy is no longer valid. IPFS: Its primary function is to alleviate the storage burden on the blockchain. Medical data preserved in IPFS will be housed in a MerkledAG to uphold data security, known as the address hash. Subsequently, the address hash is integrated into the zone chain, thereby substituting the original data. Within IPFS, the original data undergoes the SHA256 algorithm twice and then undergoes Base58 encoding, yielding a hash length of 33 Bytes. Consequently, the original medical details are

substituted with the hash address, leading to a significant reduction in the size of a block. Blockchain: Serving as the core of the solution, the blockchain represents a decentralized network of reliable nodes that ensures the synchronization and retention of medical information, thereby upholding data integrity and security. In this solution, the blockchain is constructed based on Hyperledger Fabric, with access control being enforceable through the composition of smart contracts.

Workflow of the system model

This section describes the various stages in the process of the proposed system, which mainly comprises four parts.Then each part is explained, and the specific workflow is illustrated in figure 9.

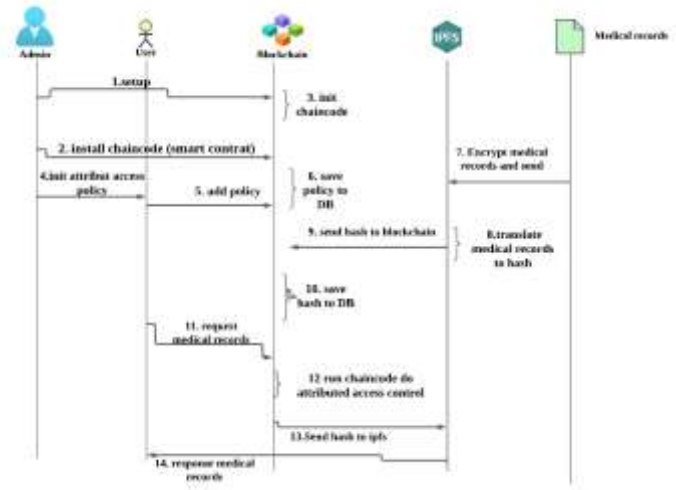

Fig 9. Workflow of system model

Part 1: The initial phase of this program involves the implementation of the Chaincode structure within the blockchain network. Nonetheless, it is imperative that these fundamental procedures are carried out by the designated user or administrator. The primary Process 1 can be delineated into a sequence of three distinct steps.

Step 1: Before creating a specific blockchain network, all network members must register the user name and certificate. Network members must register the certificate and the required certificate is issued by the CA.

CA → {Certpeer, Certorderer, Certchannel, Certuser} (6)

All peer and principal nodes are executed in Docker containers and the certificates they require must be embedded in a Docker image before being executed.

Build(conf, Cert)buid-→ Image run -→ Container (7)

Once you've set up all the peer and ordering nodes, start creating each one in a separate blockchain and ledger as {blockchain, ledger}join-→ Channel (8)

Step 2: Following the previous procedure, a rudimentary blockchain network is constructed, and the next step is to write the Chaincode in order to generate an application.

code(Fx)→ CC (9)

The Chaincode must be installed on all peer nodes by the administrator user using the Hyperledger Fabric SDK or Client.

Install SDK/Client (CC) → → Peer (10)

Step 3: After the chaincode is finished, it must be initialized by invoking the invoke method. The initialized chaincode is then saved in the container. Call upon

SDK/Client →→ Peer (11)

Part 2: The user and administrator must agree on the entire procedure and provide the applicable access control policies in this part. After the policy is developed, the administrator must save it to the blockchain.

Step 1: Based on AS, AO, AP, and AE, administrators and users establish access control policies. Select

(AS, AO, AP, AE) → ABACP through (12)

Step 2: The created access control policy is uploaded to the blockchain network by the administrator.

Upload (ABACP) → Agreement (13)

Step 3: The administrator saves the final policy values to the SDB and ledger and launches PSC to carry out actions like adding and changing policies.

PSC(ABACP) → {SDB, ledger} (14)

Part 3: In order to store medical data, this component first uploads the medical records to IPFS in order to obtain a hash address. It then saves that address to the blockchain.

Step 1: Users upload encrypted medical records to IPFS. (15) Upload (Health Information) -> IPFS

Step 2: Using its operating technique, IPFS converts medical records into a hash address. Translation of IPFS (Medical Record) to hash (16)

Send the hash address to the blockchain in step three.

Hash transmitted to blockchain (17) Step 4: Execute the smart contract RSC to save medical data to the SDB and ledger. {SDB, ledger } → Run(RSC) (18)

Part 4: There are four distinct processes in this part that make up the procedure of using attribute access control to obtain medical information.

Step 1: The user starts the process of requesting access to medical records. blcokchain → Request (19)
Step 2: The ASC contract is contacted to confirm that the user has access to the data after receiving a user request. (1, allow
(0, deny) → ASC(Request) (20)
Step 3: The blockchain sends the hash of the medical data to the IPFS if the user is granted access. Hash blockchain transmit →→ IPFS (21)
Step 4: Using the hash address as a basis, IPFS computes the medical information the user has requested. Cli → Response (Medical Record) (22)

Smart contract of system model

The core of this solution is smart contracts, which are associated with both the implementation of access control and the storage of medical data. The policy contract (PSC), access control contract (ASC), and medical record contract (RSC) are the three types of smart contracts that are available.

Policy Contract (PSC): The following ABACP manipulation techniques are offered by the PSC.

CheckPolicy(): The verification of the validity of the ABACP is required by the PSC through this specific procedure. It is essential that each ABACP includes AS, AO, AP, and AE, and compliance with all four attributes is necessary for the validation of this policy.

AddPolicy(): Prior to the execution of this method, the PSC must perform the CheckPolicy() function to confirm the legitimacy of the policy. Subsequently, only upon ensuring the policy's legality, can it be recorded in both the SDB and blockchain. The process details are delineated in Algorithm 1.

```
Algorithm 1 AddPolicy()
Input: ABACP
Output: Ok or Error
APIstub ChaincodeStub ← Invoke() ;
if CheckPolicy(ABACP) == False then
return ->Error;
end if
5: AS, AO ← ABACP
6. ABACPid ← HASHsha256(AS + AO) ;
7: err ← A APIstub.PutState(ABACPid, ABACP)
8. if err! = null then
9: return Error;
end if
return ->ok
```

DeletePolicy() will be invoked through two distinct approaches. Primarily, the administrator will initiate a call to this method for the purpose of eliminating an ABACP. Subsequently, in cases where the CheckAccess() method determines that a policy has expired, the method will be automatically triggered to eradicate the obsolete policy. This process is delineated in Algorithm 2.

```
Algorithm 2  DeletePolicy()
Input: AS, AO
Output: Ok or Error
APIstubChaincodeStub ← Invoke()
2 .PolicyID ← HASHsha256(AS + AO)
3. err ← APIstub.GetState(Id)
4.  if err! = null then
5. return Error
6. end if
7. err ← APIstub.DelState(PolicyID)
8. if err! = null then then
9. return Error
10. end if
11. return->Ok
```

When an administrator necessitates the modification of an ABACP, the UpdatePolicy() method is invoked. The alteration of an ABACP by the administrator prompts the execution of this method. The details of the modification are recorded in both the SDB and the blockchain. Furthermore, upon completion of the policy update, the AddPolicy() method is executed to reintegrate the modified policy into the blockchain.

The retrieval of information pertaining to ABACP is facilitated through the QueryPolicy() method. All policies are preserved within the state database CouchDB, which operates as a key-value pair database. Through the utilization of the properties AS or AO, the administrator is able to retrieve the specifics of the desired ABACP.

Access Control Contract (ASC): The main function of the ASC is to implement the access control function, which is to ascertain if a user's request for access control is in line with the approved access control policy. The following are the ASC techniques. CheckAccess(): As demonstrated by Algorithm 3, this technique is the foundation for the implementation of access control. In the event that the method returns a null result, it demonstrates that the request is invalid and that no policy supports it. There must be a policy that corresponds with the request if the outcome is not null. Ultimately, the request

undergoes verification through the validation of the eligible policy. If the policy's qualities, AE and AP, are both met, the request is considered to have passed the verification process.

Algorithm 3 CheckAccess()
Input: ABAC_resquet
Output: done or Error
1 . AuS, AuO, AuE ← GetAttrs(ABAC request)

2. P ← PSC.QueryPolicy(AuS, AuO)
3. if P == Null then
4. return Error();
5. end if
6. { . . . , ApP, ApE} ← P
7. if Value(ApP) == 1 && ApE.endTime >currentTime then
8. Return ok
9. end if
10. Return ->Error()

3) The Policy Contract (PSC) serves as the primary tool for housing a hash address that signifies a comprehensive medical record. Initially, the user uploads the medical record to IPFS, which subsequently generates a hash address for the record. This hash address is then transmitted to both the blockchain and SDB. The AddRecord() function is responsible for transferring the hash address from IPFS to the blockchain by inserting the key-value pair "recordId, hash" into the SDB. The specific steps are delineated in Algorithm.4.

Algorithm 4 AddRecord()
Input: Medical Record(MR)
Output: Ok or Error
APIstubChaincodeStub ← Invoke()
2.IPFShash ← IPFS(MR)
3. RecordID ← HASHsha256(MR.recordId)
4. err ← APIstub.PutState(RecordID, IPFShash)
5. if err! = null then
6. return Error
7. end if
8. Return->ok

The method DeleteRecord() initiates the deletion process by removing the hash address from the SDB, followed by the elimination of the entire medical record from the IPFS according to the recordId. The execution of UpdateRecord() involves the initial updating of medical data in the IPFS to generate a new hash address, subsequently transferring this new hash address to the SDB through the invocation of the AddRecord() method. In QueryRecord(), the primary step entails searching for the hash address of the medical record in the SDB based on the recordId, then transmitting the located hash address to IPFS for conversion into a comprehensive medical record.

Result and Discussion

This section shows the experiment and results that are set up to verify solution performance by comparison. V-A, this section presents the hardware and software tools used for experimentation. V-B, this section presents the implementation of the solution, V-C will present the experimental results.

Atmosphere of experimentation

The hardware and software tools used for experimentation are in table II

Hardware	Software
CPU I7; 2.9GHz	ubuntu 20.04
Memory 8G	Docker v10.15, v15.3
Hard disk : 512 or 1T	Docker-compose v1,24.1
	Node v12.1.0
	Golang : v1.15,1.17
	Hyperledger fabric : v1.4.6
	IPFS : v0.3.2 or v0.47

Conception and result

This section presents the structure of the environment configuration, the installation of chaincode and the use of attribute-based access control to call smart contracts.

1) The network architecture and initialization process entail a total of six network nodes, with detailed steps outlined below.
2) The initial step involves utilizing cryptogen tools to produce organization structure and identity certificates for the network.
3) Subsequently, the next step includes employing the configtxgen tool to create the Orderer's creation block, the channel's configuration transaction file, and the anchor node configuration update file for each organization.

4) Lastly, the process commences by launching the fabrics network through docker-compose, followed by utilizing client nodes to establish channels and subsequently incorporating each peer node into the channels.
Installing and upgrading Chaincode: Installing is first.
The blockchain code can be installed once the network has been initialized. The Hyperledger client node is used to install the chaincode. The chaincode is installed into each peer node sequentially by the client node. And now for instantiation.
Any peer node can be specified to instantiate the installed chaincode after it has been installed. Upgrade at last. The new chaincode must be installed before you can update the chaincode; in other words, the chaincode update is only effective on the peer node that has the new chain code installed.
System implementation: Users of Hyperledger Fabric have the option of interacting with the blockchain through a client or an SDK; in this case, the blockchain will be accessed through a client created by the SDK. These are the precise actions to take.

Step 1: The client receives a key pair from the CA node, which is kept in the wallet of the user. Step 2: After the client and peer node are connected by the administrator, the transaction can be processed or submitted.

Step 3: First the orderer node completes the sorting process, then a consensus is reached between the peer nodes, and finally the status database can be queried or updated.

If you want to add a policy, you can call the AddPolicy() method in PSC,

if you want to know whether a policy has been added successfully, you can call the QueryPolicy() method in the PSC to query the details of a policy.

this policy can be updated by calling the UpdatePolicy() method in the PSC for some reason to adapt to the new case. The DeletePolicy() method in the PSC can be used to erase a policy if it becomes invalid or if the administrator needs to compel its deletion.

Experiments are designed to validate this system's performance. We were able to

show the speed and throughput of the PSC and RSC in processing transactions subject to various simultaneous demands by setting the number of virtual clients at 200, 400, 600, and 800.

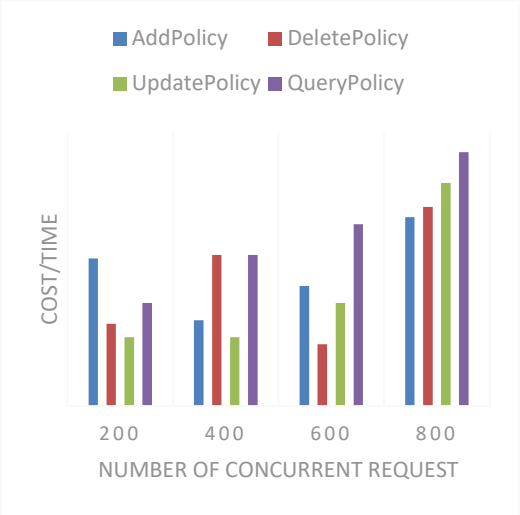

Fig 10. Time spent by RSC under different concurrent requests

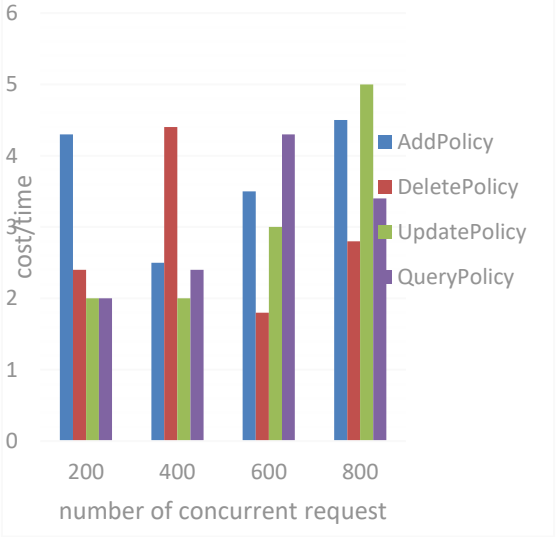

Fig 11. Time spent in PSC with different concurrent requests

Acronyme and definition

Notation	Signification
ABAC	Attribut-Based Acces Control
SDK	Software Development Kit
AS	Attribut Subject
CC	Chiancode in Hyperledger
CA	Certificate Authority
Cert	Certificate file
IPFS	Interplanetary file system
Ledger	Ledger in Hyperledger fabric
Image	Docker image
Contenair	Docker container
Fx	Function in source code
AO	Attributes Object
AE	Attributes Environment
AP	Attributes Permission
ASC	Acces smart contract
RSC	Record smart contract
Cli	Blockchain system client

Conclusion

This paper combines blockchain technology, IPFS, with an attribute-based access control model and proxy re-encryption to take full advantage of blockchain technology to break down information silos in medical data and preserve the security and privacy of medical information. In addition, the interstellar file system is used for storage to alleviate storage pressure on the blockchain. The system uses a distributed architecture to achieve fine-grained dynamic access and proxy encryption to encrypt the medical data to be stored on the interstellar file system. In conclusion, this model solves the problem of centralised storage and also provides a practical reference for related research and can give ideas to researchers. For future work, we need to examine the problem of blockchain storage without combining other technologies, and also can reimplement the system with other platforms such as Corda, hyperledger besu and sawtooth.

LIST OF ABBREVIATIONS

Abreviation	Description
ABAC	Attribut-Based Acces Control
EHR	Electronic Health Record
EMR	Electronic Medical Record
ABACP	Attribut-Based Access Control Policy
SDK	Software Development Kit
AS	Attributes Subject
CC	Chiancode in Hyperledger
CA	Certificate Authority
Cert	Certificate file
HTTP	Hyper Text Transfer Protocol
IPFS	Interplanetary file system
PKI	Public Key Infrastructure
DHT	Distributed Hash Table
Ledger	Ledger in Hyperledger fabric
ABACR	Attribute Based Access Control Request
P2P	Peer-to-Peer
Image	Docker image
Contenair	Docker container
Fx	Function in source code
AO	Attributes Object
AE	Attributes Environment
AP	Attributes of permission
ASC	Acces smart contract
RSC	Record smart contract
SDB	State database in Hyperledger fabric
PSC	Policy smart contract
SFS	Scalable File Service
Cli	Blockchain system client
DPoS	Delegate Proof Of Stack

References

[1] V. Sima, I. G. Gheorghe, J. Subi c, and D. Nancu, "Inflfluences of the industry 4.0 revolution on the human capital development and consumer behavior: A systematic review," Sustainability, vol. 12, no. 10, p. 4035, 2020.

[2] S. Schulz, R. Stegwee, and C. Chronaki, "Standards in healthcare data," Fundamentals of Clinical Data Science, pp. 19–36, 2019.

[3] M. Li, D. Han, X. Yin, H. Liu, and D. Li, "Design and implementation of an anomaly network traffific detection model integrating temporal and spatial features," Secur. Commun. Networks, vol. 2021, pp. 7 045 823:1– 7 045 823:15, 2021.
[4] D. Li, D. Han, X. Zhang, and L. Zhang, "Panoramic image mosaic technology based on sift algorithm in power monitoring," 2019 6th International Conference on Systems and Informatics (ICSAI), pp. 1329– 1333, 2019.
[5] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8770–8781, 2019.
[6] X. Zhang, L. Zhang, and D. Li, "Transmission line abnormal target detection based on machine learning yolo v3," 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 344– 348, 2019.
[7] Dagher G G, Mohler J, Milojkovic M, et al. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology [J]. Sustainable Cities and Society, 2018, 39(1): 283-297.
[8] McFarlane C, Beer M, Brown J, et al. Patientory: A Healthcare Peer-to-Peer EMR Storage Network v1 [J]. 2017.
[9] Azaria A, Ekblaw A, Vieira T, et al. MedRec: Using Blockchain for Medical Data Access and Permission Management[C]// International Conference on Open & Big Data. 2016.
[10] Ekblaw A, Azaria A, Halamka J D, et al. A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data[C]//Proceedings of IEEE open & big data conference. 2016, 13: 13.
[11] Xue T F, Fu Q C, Wang C, et al. Study on Medical Data Sharing Model Based on Blockchain[J]. Acta Automatic Sinica, 2017, 43(9): 1555-1562.
[12] U U. Health Insurance Portability and Accountability Act of 1996. Public Law 104-191[J].
[13] Ge ChunPeng. Research on Several Issues of Proxy Re-encryption [D]. Nanjing University of Aeronautics and Astronautics, 2016.
[14] Alladi, T.; Chamola, V.; Parizi, R.M.; Choo, K.K.R. Blockchain Applications for Industry 4.0 and Industrial IoT: A Review. IEEE Access 2019, 7, 176935–176951. [CrossRef]
[15] Xhafa F, Li J, Zhao G, et al. Designing cloud-based electronic health record system with attribute-based encryption [J]. Multimedia Tools and Applications, 2015, 74(10):3441-3458.
[16] F. P. Oganda, N. Lutfifiani, Q. Aini, U. Rahardja, and A. Faturahman, "Blockchain education smart courses of massive online open course using business model canvas," 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1–6, 2020.
[17] D. Li, D. Han, Z. Zheng, T.-H. Weng, H. Li, H. Liu, A. Castiglione, and K.-C. Li, "Moocschain: A blockchain-based secure storage and sharing scheme for moocs learning," Computer Standards & Interfaces, 2021.
[18] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Wang, "Blockchain enabled smart contracts: Architecture, applications, and future trends," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, pp. 2266–2277, 2019.
[19] D. Li, D. Han, and H. Liu, "Fabric-chain & chain: A blockchain-based electronic document system for supply chain fifinance," in BlockSys, 2020.
[20] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fifine-grained attribute-based data storage in cloud computing," IEEE Transactions on Services Computing, vol. 10, pp. 785–796, 2017.
[21] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," 2016 2nd International Conference on Open and Big Data (OBD), pp. 25–30, 2016.
[22] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "Medblock: Effificient and secure medical data sharing via blockchain," Journal of Medical Systems, vol. 42, pp. 1–11, 2018.
[23] A. F. da Conceic¸ao, F. S. C. da Silva, V. Rocha, A. Locoro, and J. M. Barguil, "Eletronic health records using blockchain technology," ArXiv, vol. abs/1804.10078, 2018.
[24] G. Yang and C. Li, "A design of blockchain-based architecture for the security of electronic health record (ehr) systems," 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 261–265, 2018.
[25] S. Zhang, A. Kim, D. Liu, S. C. Nuckchadyy, L. Huangy, A. Masurkary, J. Zhangy, L. P. Karnatiz, L. Mart´ınez, T. Hardjono, M. Kellis, and Z. Zhang, "Genie: A secure, transparent sharing and services platform for genetic and health data," ArXiv, vol. abs/1811.01431, 2018.
[26] J. Liu, X. L. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "Bpds: A blockchain based privacy-preserving data sharing for electronic medical records," 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, 2018.
[27] R. Qiao, X.-Y. Luo, S. Zhu, A.-D. Liu, X. Yan, and Q. xian Wang, "Dynamic autonomous cross consortium chain mechanism in e-healthcare," IEEE Journal of Biomedical and Health Informatics, vol. 24, pp. 2157– 2168, 2020.