EMRShareChain: A Privacy-Preserving EMR Sharing System Model Based on the Consortium Blockchain

Xinglong Zhang, Peng Xi, Wenjuan Liu, and Shaoliang Peng(✉)

College of Computer Science and Electronic Engineering, HuNan University, Changsha 410082, China
{xinglongzhang,slpeng}@hnu.edu.cn

Abstract. Electronic medical record (EMR) Sharing can reduce duplicate medical tests, reduce treatment costs and improve the quality of medical services. However, most medical organizations currently use the internal network to track patients but don't implement data sharing with other medical organizations. Meanwhile, the EMR is also highly vulnerable to theft and tampering by malicious attackers.

To solve the above problems, we design a privacy-preserving EMR sharing system model based on consortium blockchain, called EMRShareChain. In EMRShareChain, multiple medical organizations firstly spontaneously build up a medical consortium. The EMR generated by a patient's treatment at a medical organization is encrypted and stored securely in the cloud, and the index of the EMR is stored in the blockchain. With the above storage method, not only the risk of EMR leakage is reduced, but also the EMR cannot be tampered with arbitrarily. Then, other organizations in the medical consortium can find the patient's relevant EMR in the blockchain by keywords in the index and keyword searchable encryption, and obtain the EMR key and EMR storage address generated by the conditional proxy re-encryption after the patient's authorization to achieve access to the EMR and data sharing. In addition, an improved PBFT consensus algorithm (I-PBFT) based on comprehensive scoring mechanism is proposed to improve the blockchain block-out efficiency and stability, making it more suitable for medical scenarios. Finally, a prototype implementation of the above model is carried out based on the Hyperledger Fabric framework. The Hyperledger fabric chaincode is accessible at https://github.com/xl13455/EMRShareChain and code can be downloaded.

Keywords: EMR sharing · Consortium blockchain · Searchable encryption · Conditional proxy re-encryption

1 Introduction

With the advancement of information technology, the medical industry has shown the development trend of informatization. The emergence of EMR solves

M. S. Bansal et al. (Eds.): ISBRA 2022, LNBI 13760, pp. 343–355, 2022.
https://doi.org/10.1007/978-3-031-23198-8_31

the problems of easy loss and damage of paper medical records. However, most medical organizations currently use the internal network to keep track of their patients but don't implement data sharing with other medical organizations and increase the cost and difficulty of medical services, which brings about the phenomenon of data silos [1]. And EMR is also highly vulnerable to theft and tampering by malicious attackers, jeopardizing patient privacy and interests [2].

The rise of cloud computing has provided a solution for data sharing [3]. Medical organization uploads patients' EMRs to the cloud sharing platform in real-time, and other medical organizations in the consortium can access EMR, enabling data sharing. However, the above data sharing scheme has many problems, as the cloud is semi-trustworthy, the cloud may peek into the patient's EMR and conspire with doctors to modify the patient's EMR for profit [4]. Later, many research scholars [5,6] proposed a secure EMR sharing scheme based on cloud and attribute-based encryption, where EMR is encrypted by a set of attributes, and only users who meet the attribute permissions can access the data. However, the EMR stored in the cloud is still subject to tampering.

Fortunately, the rise of blockchain and cryptography offers the possibility of privacy protection and data sharing. Blockchain is a peer-to-peer distributed database with excellent features such as decentralization, group maintenance, tamper-evident and transparent data on the chain, making it ideal as an underlying technology for data sharing. And cryptography can protect the privacy of data, making it accessible only to authorized users.

A lot of decentralized EMR privacy protection and sharing schemes have emerged. Dubovitskaya et al. [7] used blockchain, cloud, and symmetric encryption for EMR sharing. In the scheme, EMR data is encrypted using symmetric key to ensure data privacy, and blockchain ensures that the data is difficult to tamper with. The Patient can authorize other organizations in the medical consortium symmetric key to enable data sharing. However, the scheme imposes patient to a heavy key management burden and communication overhead. Dubovitskaya et al. [8] proposed an EMR sharing scheme based on public key cryptosystem and blockchain. The scheme uses a symmetric key to encrypt the EMR and then uses the patient's public key to encrypt the symmetric key to protect data privacy. The scheme then implements data sharing by the patient encrypting the symmetric key using the doctor's public key. However, the frequent encryption and decryption work imposes a computational burden on the patient. The EMR sharing scheme based on blockchain and attribute-based encryption [9–11] stores EMR in the cloud after encryption by patient-set access policies, and stores storage address in the blockchain. Each medical organization in the medical consortium finds the relevant EMR through the blockchain, then downloads the encrypted EMR from the storage address and decrypts it using its own attribute private key. However, attribute-based encryption is deeply affected by encryption efficiency, and the change of the access policy has been a challenge.

In this paper, a privacy-preserving EMR sharing system model based on consortium blockchain is proposed. The specific contributions are as follows:

(1) A collaborative cloud-chain secure storage approach is proposed to protect data privacy and integrity.
(2) A joint-design of conjunctive-keyword searchable encryption and conditional proxy re-encryption is proposed to achieve data search and secure sharing.
(3) An improved PBFT consensus algorithm based on a comprehensive scoring mechanism is proposed to improve the performance and stability of the blockchain outgoing blocks.

The remainder of this paper is organized as follows. In Sect. 2 we briefly introduce the preliminary. In Sect. 3, we describe the architecture and workflow of the EMRShareChain. In Sect. 4, we test the performance and analyze security of EMR. Finally, we conclude the full paper in Sect. 5.

2 Preliminary

2.1 Blockchain

Blockchain is a technical solution for storing, transferring and exchanging network data through its own distributed nodes without relying on third parties. Decentralization, group maintenance, non-tamperability and transparency are the features of blockchain [12]. The current blockchains can be divided into three categories: public blockchain, private blockchain and consortium blockchain.

The openness of the consortium blockchain is between public blockchain and private blockchain and is jointly managed by multiple organizations. Only authorized users in authorized organizations can initiate and view blockchain transactions, such as Hyperledger Fabric [13]. In this paper, we build the Hyperledger Fabric consortium blockchain by different medical organizations spontaneously providing nodes and sharing data by storing the EMR index in the blockchain.

2.2 The Conditional Proxy Re-encryption

Conditional proxy re-encryption(CPRE) [14] is able to re-encrypt the eligible ciphertext under Alice's public key into ciphertext under Bob's public key with the help of a semi-trusted proxy. And proxy has no access to information about the original plaintext. A CPRE algorithm consists of the following six function.

$CPRE.KeyGen(i)$: This function generates the key pair (pk_i, sk_i) for user i.

$CPRE.ReKeyGen(sk_i, pk_i, pk_j, c)$: This function takes the private key sk_i of user i, the public keys pk_i of user i and pk_j of user j, a condition value c as input, and generates a re-encryption key $rk_{i,j}$.

$CPRE.Encrypt(pk_i, m, c)$: This function takes the public key pk_i of user i, the plaintext m and a conditional value c as input and generates a ciphertext c_i as output.

$CPRE.reEncrypt(rk_{i,j}, c_i)$: This function takes the re-encryption key $rk_{i,j}$, the ciphertext c_i as input, and generates a re-encrypted ciphertext d_j as output.

$CPRE.Decrypt(sk_i, c_i)$: This function takes the private key sk_i of a user i, a ciphertext c_i as input, and generates a plaintext m as output.

$CPRE.ReDecrypt(sk_j, d_i)$: This function takes the private key sk_j of a user j, a re-encrypted ciphertext d_i as input, and generates a plaintext m as output.

2.3 The Public-Key Encryption with Conjunctive Keyword Search

The public key encryption with conjunctive keyword search (PKSE) [15] enables data users to search files containing several keywords over a public key encryption setting. The scheme consists of the following four functions.

$PKSE.KeyGen(i)$: This function generates key pair (pk_i, sk_i) for user i.

$PKSE.keywordIndex(pk_i, W)$: This function takes the public key pk_i of user i and a keyword set W as input, and generates a encrypted keyword CW.

$PKSE.Trapdoor(sk_i, Q)$: This function takes the private key sk_i of user i and a keyword query set Q as input, and generates a search trapdoor CT.

$PKSE.Search(CW, CT)$: This function takes the encrypted keyword set CW and a keyword search trapdoor CT as input, and generates a value. If CT is included in CW, the server outputs 1, otherwise 0.

3 EMRShareChain: The Proposed System Model

3.1 System Architecture

The EMRShareChain adopts a three-layer architecture, including data generation layer, data storage layer, and data sharing layer, as shown in Fig. 1. The functional description of each layer is specified as follows.

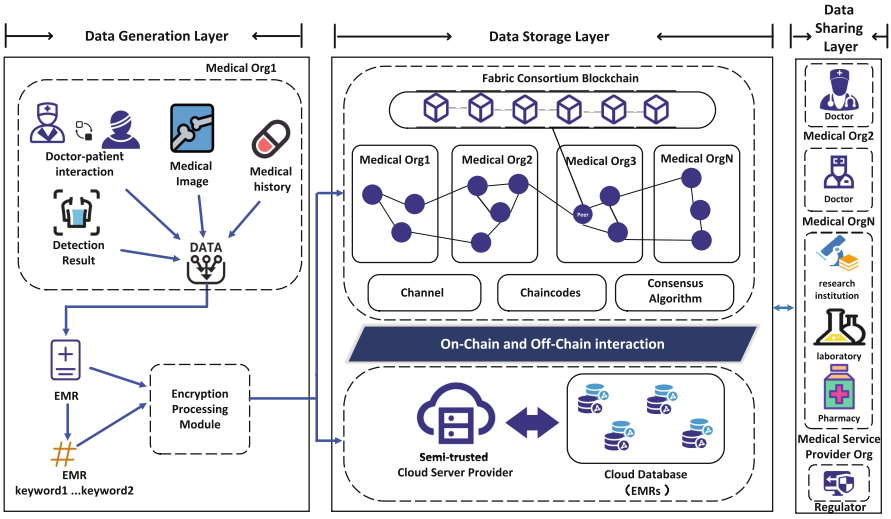

Fig. 1. The Architecture Of EMRShareChain system model

(1) **Data generation layer.** In this layer, an original EMR is generated when a patient is treated at one of the medical organizations in the medical consortium. Since the EMR contains much sensitive information, the patient has the right to entrust the doctor to encrypt the EMR using a symmetric key.
(2) **Data storage layer.** In this layer, we use cloud service provider (CSP) and blockchain to store EMR and the EMR index, respectively.

– CSP: CSP is responsible for storing the encrypted EMR and returning the corresponding storage address after successful storage.
– Consortium blockchain: The blockchain is responsible for storing index to realize data sharing, where the index contains id, digital digest, digital signature, encrypted keywords, encrypted symmetric key, and encrypted storage address.

(3) **Data sharing layer.** At this layer, any organization or user in the medical consortium authorized by the patient can search the EMR on the blockchain and obtain the re-encrypted symmetric key and the re-encrypted storage address. The organization or user can use the private key to decrypt the ciphertext and obtain the EMR to provide better care to patients or conduct scientific research.

3.2 System Workflow

In the following, we will describe the workflow of EMRShareChain. The whole workflow is divided into four phases: system initialization phase, data generation phase, data processing and storage phase, and data search and sharing phase.

System Initialization Phase. During this phase, the purpose is to build the blockchain network and complete user registration. (i) Regarding the blockchain construction, consortium blockchain includes three types of nodes: CA node, peer nodes, and orderer nodes. (ii) Regarding user registration, each user of the organization needs to provide real identity information for registration and will get the key pairs and digital certificates for joining the blockchain network.

CA Join Phase. The CA is an important part of the consortium blockchain and is managed by government personnel. Any organization and user joining the blockchain must have key pairs and digital certificates issued by the CA.

Orderer Join Phase. The orderer service is responsible for sorting and packaging blockchain transactions. Each node of orderer service is provided by the medical organization M_j. The M_j provides the real identity information RID to the CA for registration, and then CA generates the identity key pair $(sk_{iden(M_j)}, pk_{iden(M_j)})$, TLS key pair $(sk_{tls(M_j)}, pk_{tls(M_j)})$, and identity digital certificate $x509Cert_{iden(M_j)}$ and TLS digital certificate $x509Cert_{tls(M_j)}$ needed to build the orderer node for it. Among them, the identity key is used to indicate the identity of the organization or user, and tls key is used to ensure the security of the communication. Then, M_j sets up the orderer service.

Peer Join Phase. All medical organizations need to provide server as peer node to maintain the blockchain ledger. Similarly, the medical organization M_j needs to obtain key pairs and digital certificates required to build the peer node. Then M_j sets up the peer node service. Further, the peer node needs to join the corresponding blockchain channel and install the chaincode (cc).

User Join Phase. The user of all organization needs to go through verification when they join the blockchain network. Taking the patient user of the medical organization as an example, the patient P_i provides real identity information RID to CA. Then, CA generates the key pair and digital certificate for P_i.

In addition to this, the patient P_i also needs to generate two key pairs locally. Specifically, encryption key pair $(sk_{enc(P_i)}, pk_{enc(P_i)})$ are used to implement encryption and decryption of the data, and search key pair $(sk_{search(P_i)}, pk_{search(P_i)})$ is used to implement encryption and search of keywords.

Data Generation Phase. During this phase, the patient P_i is treated by doctor D_i at one medical organization M_j in the medical consortium, and the interaction data, test results and medical images is collected and formed into an EMR.

Data Processing and Storage Phase. During this phase, the EMR generated by patient P_i will be encrypted and uploaded to CSP. The doctor D_i first encrypts the EMR using a symmetric key Key to generate the encrypted data EMR_e, then uploads data EMR_e to the CSP and obtains the corresponding storage address Url.

The doctor D_i then encrypts the keyword $Keyword$ of the EMR, the symmetric key Key used to encrypt the EMR, and the storage address Url using the public key of patient P_i and the conditional value c to generate the encrypted keyword $Keyword_e$, the encrypted symmetric key Key_e and the encrypted storage address Url_e.

After encryption operations are completed, the doctor D_i further calculates the digital digest $digest_{EMR_e}$ of the encrypted EMR and generates digital signature $sign_{EMR_e}$ using the own identity private key $sk_{iden(D_i)}$.

Finally, The doctor D_i then feedbacks the information generated during the above operations to the patient P_i. After receiving the feedback information, the patient P_i extracts the necessary information to build the EMR index $EMRIndexEntity$ as shown in Fig. 2 and uploads it to the blockchain.

Data Search and Sharing Phase. During this phase, data users who wish to share access the patient's EMR to provide better medical service or conduct research. Here we take a doctor as an example, when the patient P_i is referred to another medical institution M_{j+1}, the doctor D_i' needs to obtain the patient's previous EMR to assist in diagnosis.

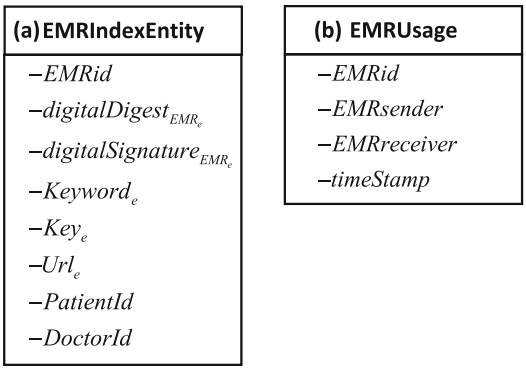

Fig. 2. The EMRIndexEntity and EMRUsage data structure

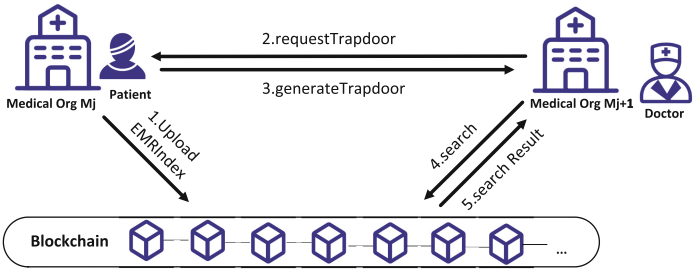

Fig. 3. The data search phase

The doctor D_i' first needs to obtain a search trapdoor *trapdoor* generated by the patient P_i based on search keywords Sw. The doctor then invokes the search chaincode with the search trapdoor *trapdoor* as an input parameter, and the search chaincode will return the matching EMR ID to the doctor D_i'. And the flow of the search phase is shown in Fig. 3.

Because EMRs are private, any user who wants to actually access the EMR plaintext must be authorized by the patient P_i.

The doctor first needs to generate a *dataAccessRequest* based on information such as the EMRid and the public encryption key of the doctor, and sends it to the patient. If the patient agrees to the access request, the patient will generates a re-encryption key *reEncKey* using his encryption private key $sk_{enc_{P_i}}$ and the doctor's encryption public key $pk_{enc_{(D_i')}}$ along with the conditional values c. Then, the patient sends the EMRid, re-encryption key to the master node of the consortium blockchain. The master node performs the re-encryption work, and sends the re-encrypted key Key_{reEnc} and re-encrypted storage address Url_{reEnc} to the doctor D_i'.

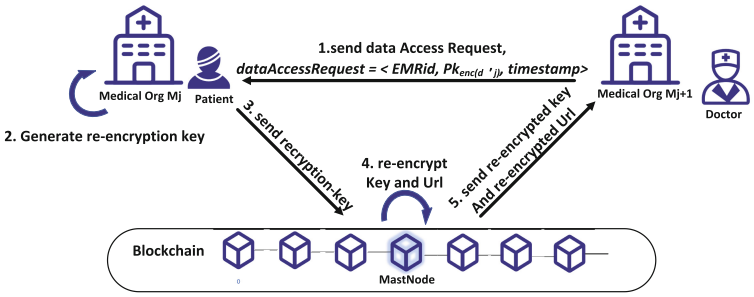

Fig. 4. The data sharing phase

The doctor D_i' can decrypt the re-encrypted storage address and then download the EMR from the CSP. At this point, the data sharing phase is completed. And the flow of the sharing phase is shown in Fig. 4.

At the same time, the patient needs to construct the data usage record $EMRUsage$ as shown in the Fig. 2 to record the details of each data usage, and then the patient uploads it $EMRUsage$ to the blockchain network. The patient can then query the data usage through the blockchain.

I-PBFT (Improved PBFT Consensus Mechanism). PBFT algorithm, as a state machine copy replication algorithm, can provide $(n-1)/3$ fault tolerance (n is the total number of nodes in the blockchain network). The mechanism can not only start and run on fewer nodes but also does not require a lot of computing power to maintain. Considering that the node number of medical organizations is small, the PBFT is more suitable for medical scenarios.

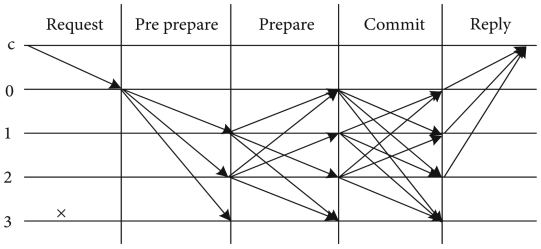

Fig. 5. The principle of PBFT consensus mechanism

This model adopts the Fabric consortium blockchain and uses the PBFT consensus algorithm, in which the nodes in PBFT are equivalent to the orderer nodes in the Fabric blockchain, and the principle of PBFT mechanism is shown in Fig. 5. In PBFT, the nodes are divided into master and slave nodes. The master node is responsible for sorting the transactions and distributing them to the slave

nodes, and generating blocks after the nodes reach consensus. However, in the original PBFT, the master node is randomly selected, which will cause frequent replacement of master node when computationally inadequate or potentially malicious medical organization act as the master node, affecting the blockchain's block-out performance and stability. Therefore, we improve on the original PBFT by changing the selection method of the master node from randomly selecting medical organization to serving as the one with the high overall score.

The score of the medical organization nodes is shown in Eq. 1, and the final score ($finalScore$) of each node is calculated based on the hardware level score ($hardwareScore$) and the consensus performance score ($consensusScore$).

$$finalScore = 0.2 * hardwareScore + 0.8 * consensusScore \tag{1}$$

The hardware level score is calculated by combining the CPU processing frequency score ($h1$), CPU core count score ($h2$), memory score ($h3$), and disk score ($h4$), and bandwidth score ($h5$), as shown in Eq. 2. The k_i of Eq. 2 indicates the different weighting factors. And it should be noted that the hardware level score is normalized.

$$hardwareScore = k1 * h1 + k2 * h2 + k3 * h3 + k4 * h4 + k5 * h5 \tag{2}$$

The node consensus performance score is calculated from the number of consensus successes (cs), consensus failures (cf), consensus successes as master node (mcs), and consensus failures as master node (mcf) in the past n consensus times by the node, as shown in the Eq. 3. It is worth noting that the node consensus level scores are normalized.

$$consensusScore = e^{\frac{(cs+mcs)-(\alpha \times cf+\beta \times mcf)}{n}} \tag{3}$$

4 Model Comparison and Performance Test

4.1 Model Comparison

We compare EMRShareChain system model with other sharing schemes in six aspects: whether it is based on blockchain, whether it requires tokens, power requirement, consensus algorithm, storage method, whether it achieves data privacy protection and fine-grained access control. For the convenience of representation, the six aspects are represented by F1, F2, F3, F4, F5, and F6. The comparison of different schemes is shown in Table 1.

Table 1. Comparison of EMRShareChain system with other systems or schemes.

Items	[5]	[6]	[7]	[8]	[9]	[10]	[11]	Ours
F1	F	F	T	T	T	T	T	T
F2	–	–	F	F	F	F	T	F
F3	–	–	Low	Low	Low	Low	High	Low
F4	–	–	PBFT	PBFT	PBFT	-	EOS	I-PBFT
F5	Cloud	Cloud	Cloud	Cloud	IPFS	Cloud	IPFS	Cloud
F6	T	T	T	T	T	T	T	T

Our scheme uses blockchain and cloud collaboration for reliable data storage. However, Zhang and Wang's schemes [5,6] use the cloud alone for storage and does not guarantee data integrity. Dubovitskaya's schemes [7,8] use the original PBFT consensus algorithm, and Liu [9] also uses the default PBFT consensus algorithm. Gao's scheme [11] uses the EOS consensus algorithm. Compared with the above schemes, our scheme uses an improved PBFT consensus algorithm (I-PBFT) to improve the efficiency and stability of blockchain out blocks. Moreover, our scheme uses consortium blockchain, which not only does not require tokens but also does not require strong power support. All schemes use encryption algorithms for data privacy protection and access control to ensure that only authorized users and institutions can share access to EMR data, but in our scheme, patients only need to generate a re-encryption key to complete access authorization operations, greatly reducing key management and communication overhead.

4.2 Performance Evaluation

The performance test is carried out to evaluate the efficiency of data processing and sharing. In order to simulate the real environment, the blockchain network CA node, orderer nodes and peer nodes are simulated by deploying docker containers in the host. The host computer is Intel(R) Xeon(R) CPU E78890 v3 and the operating system is Ubuntu20.

In the data processing and storage phase, we perform the data encryption operation (T_{de}), the digital digest calculation operation of encrypted data (T_{ddc}), the digital signature calculation operation of encrypted data (T_{dsc}), the encrypted data upload operation (T_{edu}), the keyword encryption operation (T_{kwe}), the storage address encryption operation (T_{sde}), key encryption operation (T_{ke}), EMR index upload operation (T_{eiu}) and EMR index query operation (T_{eiq}) are tested for their time cost overhead (ms). And the test results are shown in Table 2.

Table 2. The time cost overhead of operations in data processing and storage phase

File size	T_{de}	T_{ddc}	T_{dsc}	T_{edu}	T_{kwe}	T_{sde}	T_{ke}	T_{eic}	T_{eiu}	T_{eiq}
64KB	15	1	5	5	150	135	115	1	51	44
256KB	18	2	5	7	150	135	115	1	51	44
1024KB	28	6	5	12	150	135	115	1	51	44

In the data search and sharing phase, we test the time cost overhead of the search trapdoor generation operation (T_{stg}), data search operation (T_{ds}), the re-encryption key generation operation (T_{rkg}), encryption key and storage address re-encryption operation (T_{ksr}), encryption key and storage address decryption operation (T_{ksd}), encrypted data download operation (T_{edd}), encrypted data decryption operation (T_{edd2}), EMR usage record construction operation (T_{euc}), EMR usage record upload operation (T_{euu}), and EMR usage record query operation (T_{euq}). And the test results are shown in Table 3.

Table 3. The time cost overhead of operations in the data search and sharing phase

File size	T_{stg}	T_{ds}	T_{rkg}	T_{ksr}	T_{ksd}	T_{edd}	T_{edd2}	T_{euc}	T_{euu}	T_{euq}
64 KB	12	72	16	285	64	1114	8	1	35	34
256 KB	12	72	16	285	64	1333	11	1	35	34
1024 KB	12	72	16	285	64	2745	17	1	35	34

4.3 Security Analysis

Data Confidentiality. EMR is encrypted with symmetric key and stored in the cloud server, and the symmetric key and storage address are stored on the blockchain after encryption. Without the patient's private key, the attacker cannot get the plaintext information of EMR, which effectively ensures the confidentiality of EMR data.

Data Integrity. The digital digest of the encrypted EMR is calculated and stored in the blockchain. Because the data on the blockchain is difficult to tamper with, the digital digest stored on the blockchain can be used to verify the integrity of the data.

Data Authenticity. The encrypted EMR is signed by the doctor and stored permanently in the blockchain, which binds the authenticity of the EMR to the doctor's reputation and ensures data authenticity.

5 Conclusion

In this paper, a privacy-preserving EMR sharing system model based on consortium blockchain is proposed, called EMRShareChain. In EMRShareChain, the EMR generated by a patient after a visit to a medical organization is stored encrypted in the cloud, and the EMR index is stored in the tamper-proof consortium blockchain. With the above joint storage method of cloud and blockchain, the problem of EMR data being easily leaked and maliciously tampered with is solved. Meanwhile, the joint-design of conjunctive-keyword searchable encryption and conditional proxy re-encryption enables organizations in a medical consortium to quickly retrieve the relevant EMR on the blockchain and achieve secure shared access to EMR with patient authorization. Further, the improved PBFT blockchain consensus algorithm proposed in conjunction with the actual medical scenario effectively improves the model performance and stability. The implementation of the proposed EMRShareChain will enable secure sharing of EMR data between different medical organizations, thus greatly facilitating cross-institution treatment and effectively providing easy access to data for scientific research.

Acknowledgements. This work was supported by National Key R&D Program of China 2022YFC3400404; NSFC Grants U19A2067; Science Foundation for Distinguished Young Scholars of Hunan Province (2020JJ2009); Science Foundation of Changsha Z202069420652, kq2004010; JZ20195242029, JH20199142034; The Funds of State Key Laboratory of Chemo/Biosensing and Chemometrics, the National Supercomputing Center in Changsha (http://nscc.hnu.edu.cn/), and Peng Cheng Lab.

References

1. Liu, J., Li, X., Ye, L., Zhang, H., Du, X., Guizani, M.: BPDS: a blockchain based privacy-preserving data sharing for electronic medical records. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2018)
2. Zhao, Y., et al.: Research on electronic medical record access control based on blockchain. Int. J. Distrib. Sens. Netw. **15**(11), 1550147719889330 (2019)
3. Agyekum, K.O.B.O., Xia, Q., Sifah, E.B., Cobblah, C.N.A., Xia, H., Gao, J.: A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain. IEEE Syst. J. **16**(1), 1685–1696 (2021)
4. Kelbert, F., et al.: Securecloud: secure big data processing in untrusted clouds. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 282–285. IEEE (2017)
5. Zhang, A., Wang, X., Ye, X., Xie, X.: Lightweight and fine-grained access control for cloud-fog-based electronic medical record sharing systems. Int. J. Commun. Syst. **34**(13), e4909 (2021)
6. Wang, T., Zhou, Y., Ma, H., Zhang, R.: Enhanced dual-policy attribute-based encryption for secure data sharing in the cloud. Secur. Commun. Netw. 2022 (2022)
7. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and trustable electronic medical records sharing using blockchain. In: AMIA Annual Symposium Proceedings, vol. 2017, p. 650. American Medical Informatics Association (2017)

8. Dubovitskaya, A., Baig, F., Xu, Z., Shukla, R., Zambani, P.S., Swaminathan, A., Jahangir, M.M., Chowdhry, K., Lachhani, R., Idnani, N., et al.: ACTION-EHR: patient-centric blockchain-based electronic health record data management for cancer care. J. Med. Internet Res. **22**(8), e13598 (2020)
9. Liu, J., Wu, M., Sun, R., Du, X., Guizani, M.: BMDS: a blockchain-based medical data sharing scheme with attribute-based searchable encryption. In: ICC 2021-IEEE International Conference on Communications, pp. 1–6. IEEE (2021)
10. Zhang, L., Peng, M., Wang, W., Su, Y., Cui, S., Kim, S.: Secure and efficient data storage and sharing scheme based on double blockchain (2021)
11. Gao, H., Ma, Z., Luo, S., Xu, Y., Wu, Z.: BSSPD: a blockchain-based security sharing scheme for personal data with fine-grained access control. Wirel. Commun. Mob. Comput. 2021 (2021)
12. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
13. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp. 1–15 (2018)
14. Paul, A., Selvi, S.S.D., Rangan, C.P.: A provably secure conditional proxy re-encryption scheme without pairing. Cryptology ePrint Archive (2019)
15. Farràs, O., Ribes-González, J.: Provably secure public-key encryption with conjunctive and subset keyword search. Int. J. Inf. Secur. **18**(5), 533–548 (2019). https://doi.org/10.1007/s10207-018-00426-7