Journal of Information Security and Applications 61 (2021) 102952

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

An improved convolution Merkle tree-based blockchain electronic medical record secure storage scheme

Hegui Zhu [a,*], Yujia Guo [a], Libo Zhang [b]

[a] College of Sciences, Northeastern University, Shenyang, 110819, China
[b] Department of Radiology, The General Hospital of Northern Theater Command PLA, Shenyang 110016, China

ARTICLE INFO

Keywords:
Electronic medical record
Blockchain
Convolution operation
Improved convolution Merkle tree

ABSTRACT

Presently, more and more electronic medical records (EMR) are used to replace traditional recording methods. However, there has potential safety hazard in the transmission of EMR because of the personal privacy disclosure. So, how to store, transmit and share EMR effectively and securely has become a research hotspot. In this paper, we propose an improved Merkle tree based-blockchain EMR storage scheme. The hallmark of the proposed scheme is that we employ the convolutional layer structure to replace the original binary tree structure in the proposed convolution Merkle tree, which can improve the efficiency effectively. Experiments show that the number of stored nodes has decreased significantly with the same amount of input data, and the layers number of the improved convolution Merkle tree and hash calculated amount are all reduced dramatically. The security and efficiency analysis also illustrate that the proposed scheme can provide a reliable choice for the further development of data storage security in the future.

1. Introduction

Blockchain is a new application method which consist of distributed data storage, point-to-point transmission, consensus mechanism, cryptographic technology and other computer technologies. Since Nakamoto et al. first proposed the concept of bitcoin in 2008 [1]. Blockchain as the core technology of bitcoin has gradually become well known. Blockchain technology [2–4] has the advantages of decentralization, distributed storage and non-tampering, which can provide higher security and advantages in protecting patient privacy and authorized access [5,6]. Many researchers have considered applying blockchain technology to data storage, transmission and privacy protection in recent years [7,8]. Xue et al. [9] proposed a blockchain-based medical data sharing scheme, which could realize the data access and sharing for patients and doctors with high security. Zhang et al. [10] employed a blockchain-based secure and privacy-preserving personal health information sharing (BSPP) model, which could help to improve the accuracy of disease diagnosis. Kosba et al. [11] presented a decentralized smart contract system that did not store financial transactions in the clear on the blockchain, which retained transactional privacy and automatically generated an efficient cryptographic protocol using cryptographic primitives such as zero-knowledge proofs. Dagher et al. [12] utilized smart contracts in an Ethereum-based blockchain for heightened access control and obfuscation of data, and employed advanced cryptographic techniques for preserving the privacy of patients and sensitive information. Wu et al. [13] provided a new and more effective national EMR-exchanging architecture by adopting the international XDS integration profile, which can be transferred on other cloud platforms such as Google, Amazon, and so on.

In addition, electronic medical records (EMR) contain a large number of users' personal privacy information such as the user's basic information, examination reports, doctors' advice, some ultrasound image and etc, and a huge amount of medical data is generated every day. Therefore, how to store these information efficiently and safely has become more and more important. Since cloud computing has high reliability, high performance and low cost in storing and managing medical data, It had provided a good solution to the problem of large amount of stored information but not enough capacity. For example, Cheng et al. [14] proposed a method to divide big data into a series of ordered data columns and then placed these data columns on various cloud storage servers to ensured the data security. Ren et al. [15] analyzed the requirements of the digital continuity of electronic record according to the data quality theory, then constructed the first technology framework of the digital continuity guarantee to ensure the consistency, completeness and timeliness of electronic record. Zhang et al. [16] proposed a novel privacy-preserving cloud storage scheme for electronic health records based on Shamir's Secret Sharing, which

[*] Corresponding author.
E-mail address: zhuhegui@mail.neu.edu.cn (H. Zhu).

https://doi.org/10.1016/j.jisa.2021.102952

Available online 5 August 2021

could drastically boost the efficiency and performance. Liu et al. [17] proposed a fine-grained EHR access control scheme, which allowed access policies encoded in linear secret sharing schemes and can be suitable for mobile cloud computing.

However, cloud storage also has its drawbacks, if the information is stored directly in the cloud server, the private data will be leaked once the cloud server is maliciously attacked [16,17].The unique security features of blockchain technology(immutability and decentralization) can improve the security of data storage and transmission. So, some scholars have studied the joint application of blockchain and cloud technology. Cao et al. [18] provided a secure cloud-assisted eHealth system to protect outsourced EHRs from illegal modification with blockchain technology. Xia et al. [19] employed a blockchain-based MeDShare system which provided data provenance, auditing, and control for shared medical big data in cloud repositories among trustless environment. Furthermore, blockchain technology also was applied to data storage model in smart homes under multiple cloud providers. Ren et al. [20] proposed an identity-based proxy aggregate signature (IBPAS) scheme to improve the efficiency of signature verification, which could compress the storage space and reduce the communication bandwidth.

Furthermore, because Blockchain and cloud technology can reduce costs, improve efficiency and enhance security, and it also has a significant application in the Internet environment, smart homes, smart grids etc. For example, Ren et al. [21] introduced an innovative method DCOMB method to build blockchain-based IoT data query model, which implemented queries with mining hash calculation and improved the interoperability of data and the versatility of the IoT database system. Bordela et al. [22] defined a blockchain based data-centric solution network for trust in IoT scenarios, which included a mathematical formalization and the concepts of Chain of Custody and Warranty level. Jiang et al. [23] constructed a scalable and lightweight blockchain WLAN mesh network and solve the extra overhead incurred by the mesh node.

As we know, Merkle tree is a crucial part in blockchain technology. It does not need to download all the transaction data to verify whether a transaction has been approved by the whole network. The traditional Merkel tree structure was first proposed by Ralph Merkle in his patent [24], which was originally applied to the digital signature problem in message verification [25]. Afterwards, some scholars proposed improved schemes to improve the traversal speed of Merkle tree. In 1999, Christopher et al. [26] applied the Merkle tree to batch signatures, which took the full advantage of Merkel tree and only required a single signing operation to sign a large amount of data. In 2000, Wang et al. [27] combined the Merkle tree structure with the basic digital signature technology RSA. Michael Szydlo [28] used the method of calculating only the most urgent nodes hash value to reduce the node hash value stored at one time, which reduced the amount of the hash function value calculation. Although Merkle tree structure has outstanding advantages, its linear structure and a large number of hash operations make the processing speed is not very acceptable. Moreover, the binary tree can only reduce the amount of data by half at most each time, the value of each node in each layer of the structure also needs to be stored, and resulting a large amount of data storage.

In order to overcome the above problems, this paper aims to propose a blockchain and improved Merkel tree-based secure storage access model for health data, where we use the convolution layer structure to replace the original Merkle tree structure. This operation can reduce the number of intermediate nodes, hash calculation times and Merkle tree layers. Firstly, we propose a hash encryption function to encrypt the health data. Secondly, we generate the Merkle root value by the way similar to the forward propagation of the convolutional layer in a convolutional neural network (CNN). Thirdly, we store each node value of the Merkle tree. Furthermore, we pack these generated values together in a block including the Merkle root value, the current time, previous hash and the index of data as blocks head, and the Merkle tree as blocks body. Then, a blockchain is formed by connecting these blocks in order.

The rest of this paper is organized as follows. Section 2 introduces some preliminary knowledge. In Section 3, we propose the convolutional layer operation-based improved Merkle tree structure in detail. In Section 4, we give the blockchain-based EMR secure storage model, and the analysis and experimental results are discussed in Section 5. Finally, Section 6 makes a brief conclusion of this work.

2. Related work

In this section, we will introduce some preliminary knowledge used in the proposed blockchain-based electronic health data secure storage scheme.

2.1. Convolution layer structure in convolutional neural networks

Convolutional Neural Network (CNN) is a typical feed forward neural network which mainly consists of convolutional calculations, pooling operation and activation function. It is one of the popular representative algorithms of deep learning in recent years. Research on convolutional neural networks can be traced back to last century. LeNet [29] is the first convolutional neural network recognized by network. Then, in 2012, the deeper structure of CNN: AlexNet [30] appeared in the ImageNet Contest image classification task. After that, researchers further improved network performance and proposed RCNN (Region-based CNN) [31], GoogLeNet [32], VGG (Visual geometry group) [33], etc., which could effectively improve the classification and detection ability. Usually, the classic structure of CNN includes five parts: input layer, convolution layer, pooling layer, fully connected layer and the output layer. The detailed content is shown in Fig. 1. Note that the propagation methods of convolutional layers in CNNs are mainly divided into two types: forward propagation and back propagation. In this paper we only introduce the necessary forward propagation will be used in the proposed blockchain-based EMR scheme.

Forward propagation refers to the connection between the current convolutional layer and previous layer of the current layer with local connection and weight sharing, which can reduce the number of parameters significantly. Here we let W represent the convolution kernel used on the input map X, p represents the padding size of input image. Note that the convolution kernel W moves on input map X with a fixed stride s. If the size of the input map X and the convolution kernel are $M \times M$ and $n \times n$ respectively, then the size of the output map Y is $\lfloor M + 2p - n + 1 \rfloor / s \times \lfloor M + 2p - n + 1 \rfloor / s$. In order to show the detail operation of convolution calculation, we illustrate a specific convolution calculation example in Fig. 2. Just as shown in Fig. 2, the size of the input map X is 3×3 and convolution kernel size of W is 2×2. If there are no padding and stride $k = 1$, then the size of the output map Y can be 2×2. And every value in the output map Y can be calculated with the correlation operation, for example, the convolution result 67 is calculated by $4 \times 1 + 5 \times 2 + 7 \times 3 + 8 \times 4$.

2.2. Blockchain structure

Blockchain technology originated from the concept of Bitcoin, which is a distributed shared ledger database structure. Blockchain can be divided into three categories: private chain, consortium chain and public chain. In the private chain, a company or individual has absolute control of the blockchain. However, the consortium chain is managed by the members of the authorized organizations. As for the public chain, it is completely decentralized and anyone in the chain can participate in the transaction record and message query. The data information in the Blockchain is packaged into blocks by cryptographic methods, where the blocks are linked like a chain in a chronological order. In Fig. 3, we show the simple blockchain structure where the block in the blockchain contains of at least two parts: a block header

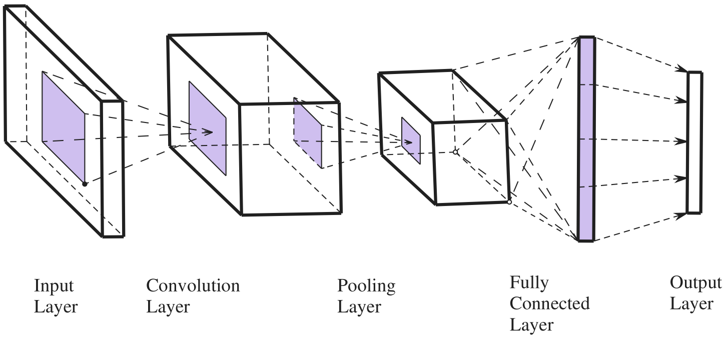


Fig. 1. Convolutional neural network.

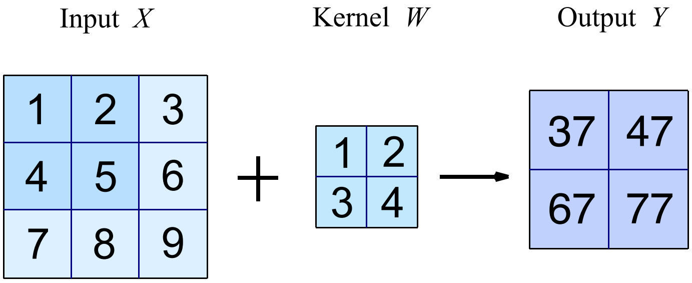


Fig. 2. Forward convolution operation in CNN.

and a block body. The block header, as the information summary of the block, is made up with the block message, the hash of the pervious block, the Merkle root and the timestamp. Note that the Genesis block has no hash of the pervious block and the hash value is 0. If an attacker wants to change the data in the block header of that block, he must start with the Genesis block. To some extend, it can avoid that malicious attackers tamper with data. When generating a block, a timestamp stored in the block header of each block to ensure the traceability of the stored data in the blockchain. Block body stores the detailed data and the Merkle tree. When the content block in the blockchain has been changed, it will result in corresponding changes to all the subsequent blocks connected to it. Consequently, we will obtain an totally different blockchain.

3. Convolutional layer operation-based improved Merkle tree structure

Due to the unique binary tree connection structure of Merkle tree, Merkle tree can ensure the security and integrity of data information. In this section, we will illustrate the improved convolutional layer operation Merkle tree structure with convolutional operation used in CNN.

3.1. Traditional Merkle tree structure

Traditional Merkle tree is a binary tree structure that stores many hash values. Fig. 4 shows a 5 layers Merkle tree structure as an example, which consists of a root node $H_{4,1}$ at the top of Merkle tree, a series of intermediate nodes $H_{3,m}, m = 1, 2$, $H_{2,j}, j = 1, 2, 3, 4$ in the middle of Merkle tree, and a group of leaf nodes $H_k, k = 1, 2, \dots, 8$ at the bottom of Merkle tree. The calculation procedure of root node $H_{4,1}$ can be shown as follows:

Firstly, let D_i be the ith input data $Data_i, i = 1, 2, \dots, 8$, $H(\cdot)$ is a hash function. Then we encrypt the input data D_i with $H(\cdot)$ to get the leaf nodes H_i, which can be calculated by

$$H_i = H(D_i), i = 1, \dots, 8. \tag{1}$$

Secondly, the next layer is donated by leaf nodes H_i, and the intermediate nodes can be computed by

$$H_{2,i} = H(H_{2i-1}, H_{2i}), i = 1, \dots, 4. \tag{2}$$

Thirdly, we repeat the above operation and generate the third layer with

$$H_{3,m} = H(H_{2,2m-1}, H_{2,2m}), m = 1, 2. \tag{3}$$

Finally, the root node result can be obtained from

$$H_{4,1} = H(H_{3,1}, H_{3,2}). \tag{4}$$

Generally, the leaf nodes generally store the hash values of a certain set of data, and the intermediate nodes store the hash values calculated by the contents of the two lower-level connected child nodes. The root node stores the last hash value of the contents of the two last intermediate nodes. Since Merkle tree has such a tight connection, then if the value of any leaf node has changed, the corresponding intermediate node value will also change. Consequently, the root node value will eventually change. For example, if $Data_4$ in Fig. 4 has changed, then the intermediate node $H_4, H_{2,2}, H_{3,1}$ will also change, and the root node $H_{4,1}$ will be very different. Therefore, if we want to verify the data of a leaf node has changed or not, we only need to compare with the two root node values of the corresponding Merkle tree.

3.2. Improved convolutional layer operation Merkle tree structure

The most important hallmark of the proposed improved Merkle tree structure is that we use convolutional layer operation replacing the traditional binary tree calculation method, which can reduce the number of intermediate nodes and layers stored in Merkle tree. At the same time, the proposed improved Merkle tree structure can decrease the number of hash calculation in generating the root node. To explain the structure in detail, we provide the proposed structure with 16 input data as an example in Fig. 5.

As illustrated in Fig. 5, D_i represents the input data need to be stored, $i = 1, \dots, 16$. S_t is the convolution kernel and the stride is 2,

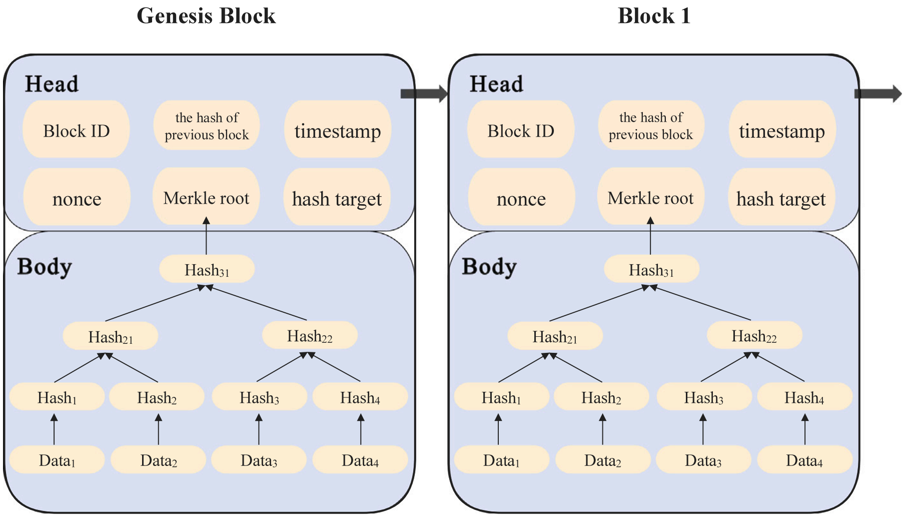


Fig. 3. General blockchain structure.

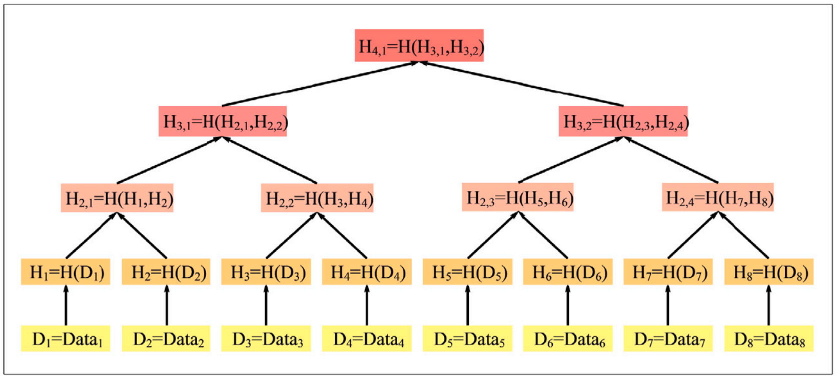


Fig. 4. Merkle tree structure.

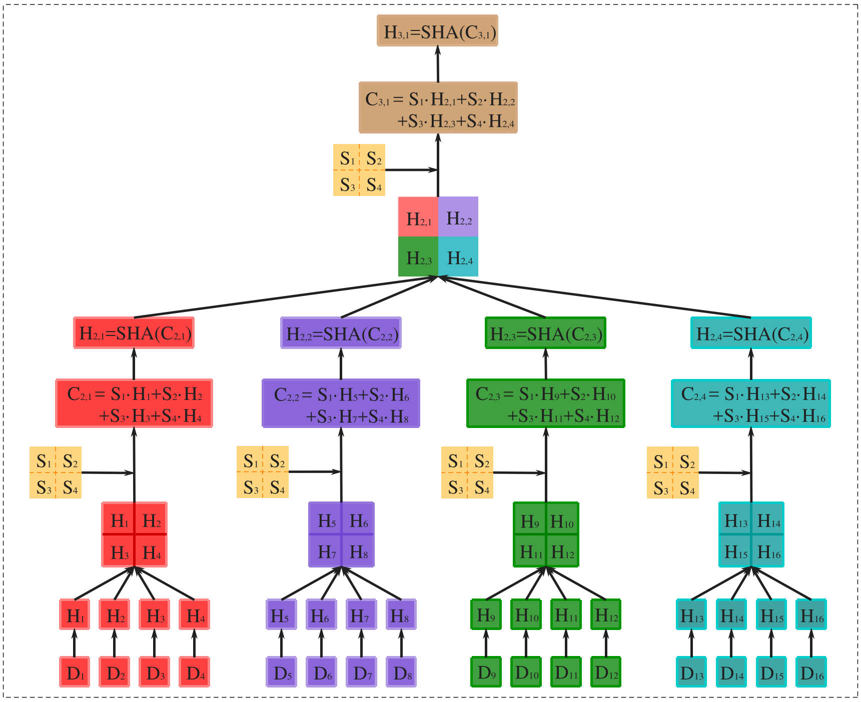

$H_{3,1}=SHA(C_{3,1})$
$C_{3,1}=S_1\cdot H_{2,1}+S_2\cdot H_{2,2}+S_3\cdot H_{2,3}+S_4\cdot H_{2,4}$
S_1 S_2 S_3 S_4
$H_{2,1}$ $H_{2,2}$ $H_{2,3}$ $H_{2,4}$
$H_{2,1}=SHA(C_{2,1})$
$H_{2,2}=SHA(C_{2,2})$
$H_{2,3}=SHA(C_{2,3})$
$H_{2,4}=SHA(C_{2,4})$
$C_{2,1}=S_1\cdot H_1+S_2\cdot H_2+S_3\cdot H_3+S_4\cdot H_4$
$C_{2,2}=S_1\cdot H_5+S_2\cdot H_6+S_3\cdot H_7+S_4\cdot H_8$
$C_{2,3}=S_1\cdot H_9+S_2\cdot H_{10}+S_3\cdot H_{11}+S_4\cdot H_{12}$
$C_{2,4}=S_1\cdot H_{13}+S_2\cdot H_{14}+S_3\cdot H_{15}+S_4\cdot H_{16}$
S_1 S_2 S_3 S_4
H_1 H_2 H_3 H_4
H_5 H_6 H_7 H_8
H_9 H_{10} H_{11} H_{12}
H_{13} H_{14} H_{15} H_{16}
D_1 D_2 D_3 D_4
D_5 D_6 D_7 D_8
D_9 D_{10} D_{11} D_{12}
D_{13} D_{14} D_{15} D_{16}

Fig. 5. Improved convolution Merkle tree structure.

$t = 1, \ldots, 4$. Here, we choose SHA256 as the hash function to encrypt D_i in the improved Merkle tree.

Now, we give the calculation process of encrypted leaf nodes generation. Firstly, we store the input Data $D_i, i = 1, \ldots, 16$ in order, if the number of input data is less than 16, then we store the integer 0 to expand to 16. Then, we encrypt the data $D_i, i = 1, \ldots, 16$ with the hash function SHA256. Finally, we store the encrypted results in $H_i, i = 1, 2, \ldots, 16$ in order. After we get the encrypted leaf nodes, we generate the hash root node which includes convolution operation on leaf nodes $H_i, i = 1, 2, \ldots, 16$ with convolution kernel $S_t, t = 1, \ldots, 4$. The detailed procedure is shown in Algorithm 1.

Algorithm 1: The generation of intermediate layers and root node with convolution operation.

```
input  : Leaf nodes $H_i, i = 1, \ldots, 16$; the convolution kernel
         $S_t, t = 1, \ldots, 4$.
output: The intermediate layers and root node with the improved
         Merkle tree.
1: divide $H_i, i = 1, \ldots, 16$ into 4 parts with the same size.
2: Input the selected convolution kernel $S_t, t = 1, \ldots, 4$.
3:for $t = 0$ to 3 do
    for $i = 1$ to 4 do
        Do the convolution operation by
        $C_{2,t+1} = \sum_{i=1}^{4} S_i \times H_{4t+i}$
    end
end
4: for $i = 1$ to 4 do
    Encrypt the sum $C_{2,i}, i = 1, \ldots, 4$ with SHA256 by
    $H_{2,i} = SHA256(C_{2,i})$
end
5: Store the encrypted results in $H_{2,k}, k = 1, \ldots, 4$ in order.
6: Obtain the third convolution layer by
   $C_{3,1} = \sum_{i=1}^{4} S_i \times H_{2,i}$
7: Encrypt $C_{3,1}$ with SHA256 by
   $H_{3,1} = SHA256(C_{3,1})$.
8: Store the encrypted result in $H_{3,1}$ as the root node.
```

From **Algorithms 1**, to deal with 16 input data, the improved Merkle tree structure only holds 4 layers, 21 hash operation and 37 nodes in total. However, if we choose the traditional Merkle tree structure, it must need 6 layers, 31 hash operations and 47 nodes. So, with the improved Merkle tree structure, the storage space and the amount of computation are all lower than the traditional Merkle tree structure. It is very useful for big input data in Blockchain-based distributed data storage and transmission.

Generally, if there are k^{2n} EMR need to be stored, in the proposed improved Merkle tree structure, firstly, we generate k^{2n} leaf nodes with hash function SHA256. Then if convolution kernel size is $k \times k$ and stride also is k. With a simple computation, there will be $n + 1$ layers and $\frac{k^{2n}-1}{k^2-1}$ nodes in the proposed improved Merkle tree structure. As for the same setting in the traditional Merkle tree structure, then the number of intermediate nodes in next layer is $\lceil \frac{k^{2n}}{2} \rceil$. In **Algorithms 2**, we give the detailed calculation of the node numbers in each layer and nodes in total.

With the above description in **Algorithms 1–2**, we take $k = 2$ as a specific example to show the difference between the improved Merkle tree structure and traditional Merkle tree structure in Table 1. As can be seen from Table 1, there are $2n + 1$ layers and $2^{2n+1} - 1$ data in the traditional Merkle tree structure. Moreover, there are $2^{2n} - 1$ hash values that need to compute before obtaining the root node value. However, the improved Merkle tree only has $n+1$ layers and $\frac{4^n-1}{3}$ times hash calculations, which stores $\frac{4^{n+1}-1}{3}$ EMR in total. Compared with the traditional structure, the improved Merkle tree structure reduces n layers, and both the number of data and hash calculation are reduced by $\frac{2^{2n+1}-2}{3}$.

Algorithm 2: Node number calculation of traditional Merkle tree structure.

```
input  : The leaf node number $k^{2n}$.
output: The layer label $i$, the node number of intermediate layer $a_i$,
         the total number of stored nodes $sum$.
if $i = 1$ then
    $a_1 = k^{2n}$
    $Sum = a_1$
end
while $a_i \neq 1$ do
    $a_{i+1} = \lceil \frac{a_i}{2} \rceil$
    $sum = sum + a_{i+1}$
    $i++$
end
```

Table 1
The difference between the improved and traditional merkle tree structure for $k = 2$.

Structure	Layers	Nodes	Hash calculation
Original	2n+1	$2^{2n+1} - 1$	$2^{2n} - 1$
Improved	$\boldsymbol{n+1}$	$\boldsymbol{\frac{4^{n+1}-1}{3}}$	$\boldsymbol{\frac{4^{n}-1}{3}}$
Difference	n	$\frac{2^{2n+1}-2}{3}$	$\frac{2^{2n+1}-2}{3}$

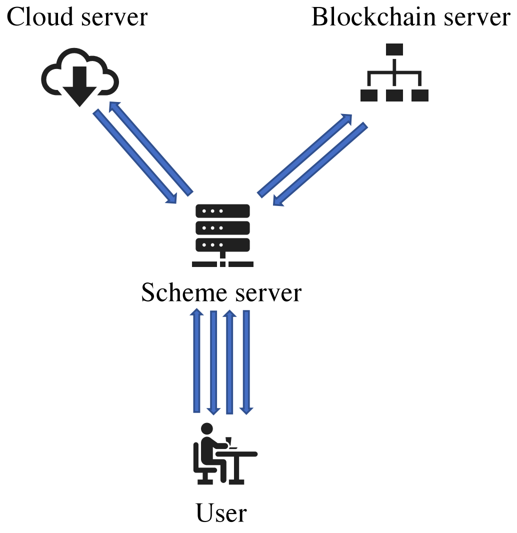


Fig. 6. The Architecture of the proposed scheme.

4. Blockchain and improved Merkle tree-based EMR secure storage scheme

In this section, in order to make it more convenient for users to record and query personal EMR on the network and ensure the security, we will employ a blockchain-based EMR secure storage scheme with improved Merkle tree structure. The detailed content is consisted of scheme setup, data storage and data query.

4.1. The proposed Architecture

The proposed blockchain-based EMR secure storage scheme consists of four entities: users, scheme server, cloud server and blockchain server, and can be shown in Fig. 6.

- *Users:* The users refer to the patients who need to upload and store EMR or other entities want to query the data. Firstly, every user should register on scheme server and get a unique ID after registration. And it should keep the ID secret and set a symmetric key K_s to encrypt data for confidentiality.

- *Scheme Server:* Scheme server is responsible for the whole EMR secure communication scheme, which generates the scheme parameters and transmits instructions between the users, blockchain server and cloud server.
- *Cloud Server:* Cloud server stores the user's encrypted ciphertext C_K and outputs the storage location and time message M.
- *Blockchain Server:* Blockchain server constructs a improved convolution based-Merkle tree with user's encrypted ciphertext C_K. Then, the encrypted message, the root value of the Merkle tree, preblock's hash value and a timestamp t in the block are also added to the block in the proposed blockchain.

4.2. Scheme setup

Scheme server generates the unique identities ID when user U registers on scheme server. User U needs to encrypt the data file F with the symmetric key K_s to generate the encrypted data file C_K by $K_s(F) \longrightarrow C_K$.

4.3. Data storage

Data storage process consists of request, storage, encryption and record. the detailed contents are shown as follows

- ***Request:*** To store the ciphertext C_K, the user U sends a request Req_S to scheme server. The request Req_S is divided into two stages.

- A storage request Req_S is submitted to scheme server: $Req_S = \{ID\|C_k\|t\}$, where t is the request time.
- Scheme server accepts the ciphertext C_K that the user U wants to store after verifying the user U's request Req_S and identity ID. Then scheme server transfers the ciphertext C_K to the blockchain server.

- ***Storage:*** After the cloud server stores ciphertext C_K, it will return the storage location and storage information M to scheme server. Then scheme server sends M to user U.
- ***Encryption:*** Having received the information M, user U encrypts M and the symmetric key K_s together to generate a tag T by $T \longleftarrow E\{M\|K_s\}$, and then sends tag T to scheme server.
- ***Record:*** User U need to record information M and ciphertext C_K on the blockchain, and the recording process consists of the following steps.

- Firstly, scheme server transmits tag T and the ciphertext C_K to blockchain server, and blockchain server generates a Merkle root value R_0 by the improved convolution based-Merkle tree structure.
- Then blockchain server packs root value R_0, tag T, hash value of the previous block and a timestamp t together in a block and adds it to the blockchain by $Block \leftarrow \{Prehash\|R_0\|T\|t\|Merkletree\}$, where t is the current time.
- Finally, blockchain server returns the index $Index_F$ of the block to the user U by scheme server, and user U receives the index $Index_F$ and stores it for the future queries.

In a word, with the operation above, a whole data storage procedure has been finished, in the following, we will illustrate the data query operation.

4.4. Data query

If the user wan to query the EMR submitted to the server, it can be realized with scheme request, blockchain query, cloud query, verification and decryption operation mentioned below.

- ***Request:*** In order to retrieve ciphertext C_K, user U submits a request Req_Q to scheme server with the following operation.

- User U requests a file F from scheme server and sends the index $Index_F$ to scheme server by $Req_Q = \{ID\|Index_F\|t\}$, where t is the request time.
- Scheme server verifies user U's identity and transmits the received index $Index_F$ to blockchain server.

- ***Blockchain Query:*** Block query can be shown in the following three steps:

- Scheme server queries the root value R_0 and the tag T from blockchain server according to the index $Index_F$.
- Then, blockchain returns the corresponding tag T and Merkle root value R_0 to the scheme server after receiving the index $Index_F$.
- Finally, scheme server returns tag T to the user U.

- ***Cloud Query:*** The user U queries the ciphertext C_K from the cloud server with tag T.

- User U decrypts tag T and obtains the storage information M with $\{M\|K_s\} = Dec(T)$, then User U sends storage location and time message M to scheme server.
- Scheme server sends storage information M to cloud server, and cloud server finds out the corresponding ciphertext C_K according to M and transmits it to scheme server.

- ***Verification:*** Scheme server generates root value R_0' of ciphertext C_K with the improved Merkle tree structure, and checkouts it with the root value R_0. If $R_0' = R_0$, then scheme server sends ciphertext C_K to the user U. Otherwise, scheme server gives the user U feedback that the file F has been tampered.
- ***Decryption:*** After user U receives the ciphertext C_K, user U decrypts it and finally gets the file F.

5. Performance analysis

In this section, we will analyze the performance of the proposed EMR storage scheme with improved convolution Merkle tree structure by security analysis and efficiency analysis.

5.1. Security analysis

In the following, we will do data integrity and privacy protection of users.

5.1.1. Tamper-proof and data integrity

The blocks in the proposed blockchain contain a timestamp, a hash of the previous block and a Merkle root. Therefore, the blocks in chronological order guarantee that the recorded data cannot be changed unless someone can simultaneously take over 51% of the computing power of the entire network. Moreover, the Merkle root is generated through the Merkle tree by all information stored in the block and block header, which also can ensure that the content of the block in the proposed blockchain network cannot be modified with unauthorized network access.

Moreover, the block header contains the calculated result of the hash function SHA256 of the previous block. SHA256 is an hash method that converts any length of data into a fixed length string. For a subtle change in input, the output of SHA256 will be very different. So, the same hash calculation result must have the same input, and there is no way to reverse the original input from the output. So, if the previous block in the proposed method has changed, and this block header also has to change. If a malicious attacker attacks the blockchain and tampers with the data in the block, the change has to be started with the header of the genesis block in order to forge a blockchain. This is a time-wasting and resource-consuming work that would be almost impossible in a large blockchain.

5.1.2. Cloud storage service and user's privacy protection

The privacy protection in the proposed scheme includes two parts: user's identity privacy and data privacy. The system generates the ID for the patient with the improved Merkel tree and SHA256, and the patient uses the ID to replace his real personal information in the medical record. Even if the attacker can obtain the specific identity information, because of the high security of improved Merkel tree and SHA256, he cannot infer the real identity of the patient. Therefore, this scheme can protect the privacy of patient's identity. Secondly, malicious attackers may steal and tamper with the data in the cloud by attacking the cloud storage server. Because the data is encrypted and stored in the cloud storage. So, the attacker cannot obtain the data content without obtaining the security key. Therefore, except for the user who has the security key can obtain the real medical records, no one else can obtain the valid medical records unless someone can obtain the user's key and ciphertext at the same time.

Moreover, an attacker must generate a parallel blockchain faster than the honest nodes to achieve a block impersonation attack. The successful attack probability of the attacker can be approximately regarded as the gambler's bankruptcy problem [34]. Assuming r is the probability of generating the next node by the honest node, a is the probability of the attacker generating the next node, and k is the number of blocks that the attacker needs to generate, then the probability of the successful attack of the attacker is

$$P = 1 - \sum_{i=0}^{k} \frac{k^i a^i e^{\frac{-ka}{r}}}{i! r^i}\left(1 - \left(\frac{a}{r}\right)^{k-i}\right). \tag{5}$$

Then we illustrate the change of P with the increasing of blocks of the blockchain in Fig. 7, where we set $a = 0.2, 0.3, 0.4$ respectively. From Fig. 7, we can conclude that the successful attack probability P goes down exponentially with the increase of blocks. Therefore, for a blockchain network with a certain number of nodes, the probability of a successful attack by an attacker is almost zero although the computing power is strong.

5.2. Efficiency analysis

5.2.1. The efficiency analysis of improved Merkle tree

To show the efficiency of the improved convolution Merkle tree structure used in the proposed blockchain-based EMR storage scheme. Firstly, we let the size of EMR data be 2^{20}, 3^{20}, and 4^{20}, respectively, then we show the nodes and layers generated with the improved Merkle tree and traditional Merkle tree structures in Table 2. From Table 2, we can conclude that the improved Merkle tree has much less total layers and nodes than the traditional Merkle regardless of the value of k, which also verifies the efficiency of the improved convolutional operation based-Merkle tree structure used in big data application.

Furthermore, to illustrate the effect more obvious, we draw the differences improved convolution Merkle tree and traditional Merkle tree in Fig. 8 with data input size is 2^{20}, 3^{20}, and 4^{20}, respectively. From 8, we can conclude that the efficiency of generating the root value of the improved convolution Merkle tree structure is significantly higher than that of the original Merkle tree. It is also obvious that with the same data input size, the improved convolution Merkle tree structure has a significant reduction in the number of layers and the amount of data calculation and storage, which also can greatly improve the efficiency.

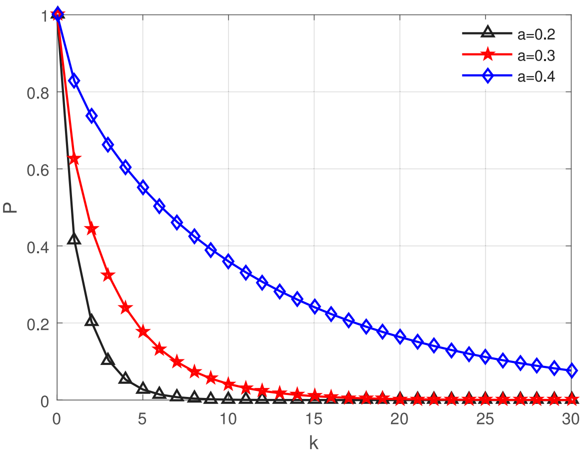


Fig. 7. The curve of the successful attack probability.

Table 2
The differences between the improved and traditional Merkle tree structures for $n = 10$.

Structure	2^{20}		3^{20}		4^{20}	
(Numbers)	layers	nodes	layers	nodes	layers	nodes
Original	21	2097151	32	6973568817	41	2199023255551
Improved	11	1398101	11	3922632451	11	1172812402961
Difference	10	699050	21	3050936366	30	1026210852590

Table 3
Structure comparison analysis.

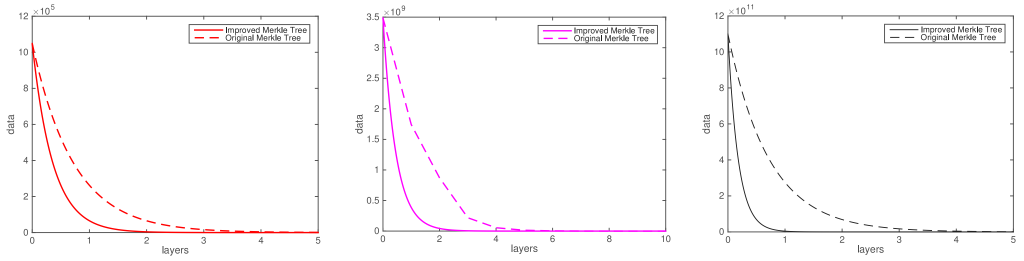
Shemes	[35]	[9]	[36]	**Ours**
blockchain	no	yes	yes	yes
consensus mechanism	–	improved DPoS	PoW	PBFT
blockchain method	–	private	consortium	consortium
control data ability	medium	strong	medium	stronger
protect privacy ability	medium	strong	medium	stronger

5.2.2. The comparison analysis of storage structure

Here we mainly evaluate the different structure of the proposed scheme with different related schemes. The detailed comparison results are shown in Table 3. As can be seen from Table 3, Ref. [35] does not apply blockchain technology, which is not secure. PoW consensus mechanism used in Ref. [36] needs to compute a specific hash value through traversal calculation, which will result in wasting resources and computing power. The private blockchain adopted in [9] has a low degree of decentralization, which is unable to meet the actual application. Furthermore, we can see that with the proposed Merkle-based blockchain consortium and PBFT consensus mechanism, the proposed method has a satisfactory performance in data controlling ability and privacy protection ability.

5.2.3. The efficiency analysis of consensus mechanism

In order to verify the execution performance of PBFT consensus mechanism used in the proposed scheme, we have compared with PoW and DPoS consensus mechanisms, and the comparison results are shown in Fig. 9. As can be seen from Fig. 9, all the execution time of the 3 mechanisms increase with the increasing of the number of transactions, while PBFT consensus mechanism of the proposed method spends minimum consumption time, and only slightly increases with the increasing of transactions number, which also can illustrate the efficiency of the proposed scheme.



(a) (b) (c)

Fig. 8. Efficiency analysis between improved convolution and traditional Merkle structure. (a) $k = 2$; (b) $k = 3$; (c) $k = 4$.

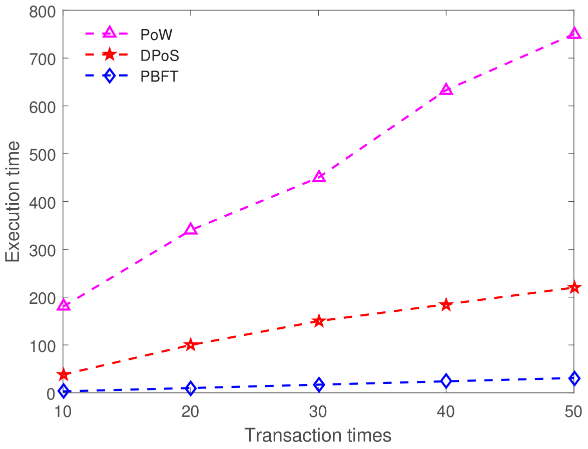


Fig. 9. The comparison results of the efficiency of consensus mechanism.

6. Conclusion

In this paper, we propose a blockchain based-EMR storage scheme with the improved convolution Merkle tree structure. Comparing with EMR storage scheme based on traditional Merkle tree structure, the proposed scheme use convolution operation to replace the original binary tree structure, which has a significant reduction in the number of layers and the amount of data calculation and storage. Furthermore, the improved Merkle tree structure is applied to the blockchain to ensure the security and convenience of data storage, which has a good performance in the EMR record and query application. It is also hoped that the structure proposed can be enlightened to some extent in other fields with the efforts of various scholars in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by the Natural science foundation of Liaoning Province, China (No. 2020–MS–080), The key research and development plan of Liaoning Province, China (2020JH/10300164).

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008, URL https://bitcoin.org/bitcoin.pdf.
[2] Underwood S. Blockchain beyond Bitcoin. Commun ACM 2016;59(11):15–7.
[3] He P, Yu G, Zhang Y-F, Bao Y-b. Survey on blockchain technology and its application prospect. Comput Sci 2017;44(4):1–7.
[4] Vujičić D, Jagodić D, Ran̄dić S. Blockchain technology, bitcoin, and Ethereum: A brief overview. In: 2018 17th international symposium Infoteh-Jahorina (Infoteh). IEEE; 2018, p. 1–6.
[5] Xu W, Wu L, Yan Y. Privacy-preserving scheme of electronic health records based on blockchain and homomorphic encryption. J Comput Res Dev 2018;55(10):2233–43.
[6] Feng Q, He D, Zeadally S, Khan MK, Kumar N. A survey on privacy protection in blockchain system. J Netw Comput Appl 2019;126:45–58.
[7] Gordon WJ, Catalini C. Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Comput Struct Biotechnol J 2018;16:224–30.
[8] Shen M, Ma B, Zhu L, Mijumbi R, Du X, Hu J. Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection. IEEE Trans Inf Forensics Secur 2017;13(4):940–53.
[9] Xue T-F, Fu Q-C, Wang C, Wang X. A medical data sharing model via blockchain. Acta Automat Sinica 2017;43(9):1555–62.
[10] Zhang A, Lin X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. J Med Syst 2018;42(8):140.
[11] Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE symposium on security and privacy. IEEE; 2016, p. 839–58.
[12] Dagher GG, Mohler J, Milojkovic M, Marella PB. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable Cities Soc 2018;39:283–97.
[13] Wu CH, Chiu RK, Yeh HM, Wang DW. Implementation of a cloud-based electronic medical record exchange system in compliance with the integrating healthcare enterprise's cross-enterprise document sharing integration profile. Int J Med Inform 2017;107:30–9.
[14] Cheng H, Rong C, Hwang K, Wang W, Li Y. Secure big data storage and sharing scheme for cloud tenants. China Commun 2015;12(6):106–15.
[15] Ren Y, Qi J, Cheng Y, Wang J, Xia J. Digital continuity guarantee approach of electronic record based on data quality theory. Comput Mater Contin 2020;63(3):1471–83.
[16] Zhang H, Yu J, Tian C, Zhao P, Xu G, Lin J. Cloud storage for electronic health records based on secret sharing with verifiable reconstruction outsourcing. IEEE Access 2018;6:40713–22.
[17] Liu Y, Zhang Y, Ling J, Liu Z. Secure and fine-grained access control on e-healthcare records in mobile cloud computing. Future Gener Comput Syst 2018;78:1020–6.
[18] Cao S, Zhang G, Liu P, Zhang X, Neri F. Cloud-assisted secure ehealth systems for tamper-proofing EHR via blockchain. Inform Sci 2019;485:427–40.
[19] Xia Q, Sifah EB, Asamoah KO, Gao J, Du X, Guizani M. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. IEEE Access 2017;5:14757–67.
[20] Ren Y, Leng Y, Qi J, Sharma PK, Tolba A. Multiple cloud storage mechanism based on blockchain in smart homes. Future Gener Comput Syst 2021;115:304–13.
[21] Ren Y, Zhu F, Sharma PK, Wang T, Wang J, Alfarraj O, Tolba A. Data query mechanism based on hash computing power of blockchain in internet of things. Sensors 2020;20(1):207–28.
[22] Bordela B, Alcarriab R, Martina D, Sánchez-Picota A. Trust provision in the internet of things using transversal blockchain networks. Intell Autom Soft Comput 2019;25(1):155–70.
[23] Jiang X, Liu M, Yang C, Liu Y, Wane R. A blockchain-based authentication protocol for WLAN mesh security access. Comput Mater Contin 2019;58(1):45–59.
[24] Merkle RC. Method of providing digital signatures. Google Patents, US Patent 4,309,569, Jan. 5 1982.
[25] Merkle RC. A certified digital signature. In: Conference on the theory and application of cryptology. Springer; 1989, p. 218–38.
[26] Pavlovski C, Boyd C. Efficient batch signature generation using tree structures. In: International workshop on cryptographic techniques and E-commerce, CrypTEC, Vol. 99. Citeseer; 1999, p. 70–7.

[27] Wang X, Hui L, Chow K, Tsang WW, Chong C, Chan H. Secure and practical tree-structure signature schemes based on discrete logarithms. In: International workshop on public key cryptography. Springer; 2000, p. 167–77.

[28] Szydlo M. Merkle tree traversal in log space and time. In: International conference on the theory and applications of cryptographic techniques. Springer; 2004, p. 541–54.

[29] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.

[30] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. COMMUN ACM 2017;60(6):84–90.

[31] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, p. 580–7.

[32] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 1–9.

[33] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, arXiv preprint arXiv:1409.1556.

[34] Zhou Z, Chen Y, Li T, Ren X, Xinyi Q. Medical data security sharing scheme based on consortium blockchain. J Appl Sci 2021;39(1):123–34.

[35] Hassan MM, Lin K, Yue X, Wan J. A multimedia healthcare data sharing approach through cloud-based body area network. Future Gener Comput Syst 2017;66:48–58.

[36] Azaria A, Ekblaw A, Vieira T, Lippman A. Medrec: using blockchain for medical data access and permission management. In: International conference on open & big data. IEEE; 2016, p. 25–30.