Blockchain-Based EMR Enhancement: Introducing PMI-Chain for Improved Medical Data Security and Privacy

Bo Cui(✉), Tianyu Mei, and Xu Liu

Inner Mongolia University, Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China
cscb@imu.edu.cn

Abstract. We have created a solution called PMI-Chain to address the challenges faced by Electronic Medical Records (EMR) management systems. These challenges include data security, patient privacy, and access regulation, which are becoming increasingly complex. PMI-Chain is a blockchain-based system that aims to tackle these issues. Traditional EMR systems rely on centralized medical data storage, exposing them to risks such as cyberattacks and data breaches and reducing patient autonomy over their medical records. PMI-Chain is a secure and confidential data-sharing platform for patients, medical institutions, and insurance companies. It uses token-based smart contracts and dual ElGamal homomorphic encryption to protect vast amounts of medical data while allowing for reliable sharing among institutions. Patients have control over their encrypted medical records, and an advanced encryption scheme keeps their information private during insurance claim processing. PMI-Chain has undergone security analysis and testing, showcasing its robust security features and exceptional stability. Additionally, compared to other available solutions, it boasts a 44% and 67% reduction in encryption and decryption time overhead.

Keywords: Blockchain · EMR · Access control · Homomorphic cryptosystem

1 Introduction

In this rapidly advancing information technology age, Electronic Medical Record (EMR) management systems [1] have become essential components of contemporary medical institutions. These systems are essential as they integrate diverse patient data elements such as diagnostic imaging, detailed clinical narratives, expert assessments, and familial genetic patterns [2]. It is important to note that electronic medical records are not only accessible to doctors but also to external entities like insurance companies. This can cause data security, patient privacy, and access regulation issues. As medical institutions adopt these technologies,

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 561, pp. 207–224, 2024.
https://doi.org/10.1007/978-3-031-54521-4_12

they face challenges that intersect with technological innovation, privacy, and compliance with regulations.

EMR systems have a central medical data storage that can be vulnerable to cyberattacks and data breaches, posing significant risks. One major concern is that patients do not have control over their medical records, which are stored in different EMR systems used by various medical institutions. This fragmentation hinders patients' direct access to and controls over their records. Additionally, patient privacy concerns frequently inhibit data sharing and interoperability between medical entities [3]. This obstruction of data flow can cause diagnostic redundancy when patients switch medical institutions, leading to an inefficient allocation of medical resources [4]. Moreover, EMR system administrators within these institutions can directly manipulate medical data, potentially facilitating fraudulent acts like insurance fraud, leading to financial losses. In addition, there is a risk of patient privacy being breached when insurance companies and other non-medical entities handle claims and review medical records. These challenges emphasize the importance of thoroughly assessing current EMR systems and developing innovative solutions that balance data accessibility, security, and patient privacy.

Numerous medical data management systems today strive to overcome previously mentioned challenges. However, despite their efforts, they often fail to meet the rigorous demands of reliability, security, and traceability crucial to medical data systems. The innovative evolution of blockchain technology emerges as a fitting solution to these issues. Blockchain's verifiability, decentralization, and immutability features [5] closely align with the pressing need for secure storage of individual health records. In a regulated blockchain system, the distributed ledger, maintained by multiple organizations, ensures medical data protection within the chain, facilitating sharing across different medical institutions. In the blockchain, every transaction within a block is given a unique timestamp and added to the chain in a way that cannot be altered. This ensures that medical data is valid and authentic, making blockchain a promising addition to EMR technology [6]. However, blockchain's inherent transparency poses concerns for the security of private data on the chain. Therefore, applying blockchain technology in the medical sector requires a careful equilibrium between data security and privacy.

Our paper presents PMI-Chain, a secure and confidential data-sharing platform for patients, medical institutions, and insurers. With PMI-Chain, we ensure fine-grained access control through token-based blockchain smart contracts. Patients can use tokens to retrieve their encrypted medical records from the smart contracts, which are encrypted using Elliptic Curve Cryptography (ECC) and symmetric keys. Patients can authenticate their records by matching the hash values stored within the blockchain network. Additionally, we integrate dual ElGamal homomorphic encryption technology with the blockchain smart contract to encrypt claim materials and amounts, enabling non-plaintext verification of health insurance claims. Another noteworthy contribution of PMI-Chain is its ability to facilitate secure data sharing between healthcare institutions, further enhancing patient care. This paper's main contributions include:

- The proposed platform is a secure and trustworthy medical information service that integrates patients, medical institutions, and insurance agencies. It is based on blockchain technology and smart contracts, ensuring medical data's authenticity. This interactive system offers a collaborative environment for studying medical-patient insurance.
- We propose a smart contract access control system that uses tokens to control access to encrypted medical data. This system allows for fine-grained control of patient information updates and improves the speed and efficiency of the access policy process. This system can securely store and share medical data between institutions.
- Our secure request system uses advanced homomorphic cryptography and improved ElGamal cryptography. This enables insurers to settle medical claims encrypted, ensuring patient privacy is protected from potential privacy violations.
- We have conducted a security analysis of PMI-Chain and found that it meets various security standards. It can also handle threats like illegal collaborations between doctors and clouds. Our tests have shown that PMI-Chain has a 44% lower encryption and a 67% lower decryption time overhead, and more stable performance than existing methods.

The rest of this paper is organized as follows. We introduce the related work in Sects. 2. We present the PMI-Chain in detail in Sect. 3. The correctness, security, privacy and efficiency analysis are described in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related Work

The digital transformation of healthcare has significantly impacted the way medical resources are allocated. Electronic Medical Records (EMR), celebrated for their instantaneous data exchange, seamless integration into healthcare workflows, and intuitive interfaces, have seen a remarkable evolution in recent years [7,8]. In their work, Li et al. [7] highlighted the semi-trusted cloud computing environments equipped with multi-level permissions. These environments seek to mitigate the risk of unauthorized access, managing an array of user role attributes via multiple permissions. The transition from conventional data centers to cloud servers, facilitated by cost-effective and secure cloud storage, has given rise to a cloud-augmented EMR architecture. This transition propels a significant surge in work efficiency. However, the involvement of third-party cloud providers may pose threats to the privacy, confidentiality, and integrity of the data. Moreover, the inherent security risks associated with cloud computing environments can lead to potential data loss. With the continuous advancement of blockchain technology, an increasing amount of research is being dedicated to its feasible application within EMR systems.

Azaria et al. [3] introduced MedRec, a prototype for distributed medical information sharing on the Ethereum platform utilizing blockchain smart contracts.

This system kept chain data in a third-party database and relied on identity verification mechanisms to ensure the accuracy and integrity of outsourced EMR. However, it could not safeguard the security of the third-party database, nor prevent unethical practices like encouraging cloud servers to alter EMR data. Cao et al. [9] proposed a solution to such issues with a secure, cloud-assisted EMR. This strategy employed an Ethereum-based blockchain to protect outsourced medical data, thereby reducing the likelihood of third-party manipulation or corruption, but it resulted in patients losing control and ownership of their medical data. Attempting to address this, Saini et al. [10] developed a distributed dynamic access control mechanism using smart contracts, empowering patients with ownership and secure sharing of their sensitive health records. However, this solution did not support inter-institutional EMR sharing. To mitigate these limitations, Fengqi et al. [2] proposed a method incorporating semi-strategy hiding and partial ciphertext CP-ABE for dynamic permission changes (SHDPCPC-CP-ABE). This strategy allowed fine-grained access control for encrypted medical data, facilitating safe exchange of health information among medical entities.

Privacy of medical data is a major concern in the healthcare industry. As blockchain technology is applied to the medical field, a balance must be struck between data security and data privacy. Research on privacy protection for blockchain-based EMR is generally divided into encryption-based and access control-based privacy protection. In the former, Benil et al. [11] devised an enhanced Elliptic Curve Cryptography (ECC) digital signature algorithm, employing Message Authentication Codes to encrypt medical data stored in the cloud for confidentiality. However, in data-sharing scenarios such as big data analysis in healthcare and insurance claim settlements, data availability must be ensured. Ding et al. [12] constructed Derepo, a private data safe storage and sharing model based on blockchain and homomorphic encryption technology, achieving privacy protection and data computability through the introduction of homomorphic encryption. On the other hand, access control-based privacy protection is exemplified by Zhang et al. [13] who proposed a blockchain-based fine-grained access control and permission revocation data sharing scheme, BDSS. It facilitated data sharing between medical institutions via fine-grained access control technology and safeguarded patient privacy through a permission revocation mechanism. However, in scenarios involving uncertain data sharers, it's impossible to predetermine access control policies. Dewangan et al. [14] proposed a data collection method based on the medical Internet of Things, generating unique tokens for users through their biometric features and random keys. This token-based design pattern enables secure access control to medical data without needing to determine data sharers in advance.

The analyses presented above underscore that data security and privacy protection remain crucial research topics for blockchain-based EMR systems. Despite various proposals, a comprehensive and secure medical data management system that accomplishes dynamic access control and privacy protection is yet to be realized. Most current systems do not fully address the data

security and privacy breach issues that may arise during the three-way interactions among hospitals, patients, and insurance companies.

3 The Proposed PMI-Chain

3.1 Overview

In this section, we elaborate on the construction of PMI-Chain, which is built upon token-based smart contracts for access control and an improved version of ElGamal encryption. As depicted in Fig. 1, PMI-Chain comprises five entities: patients, hospitals, doctors, the blockchain, and smart contracts.

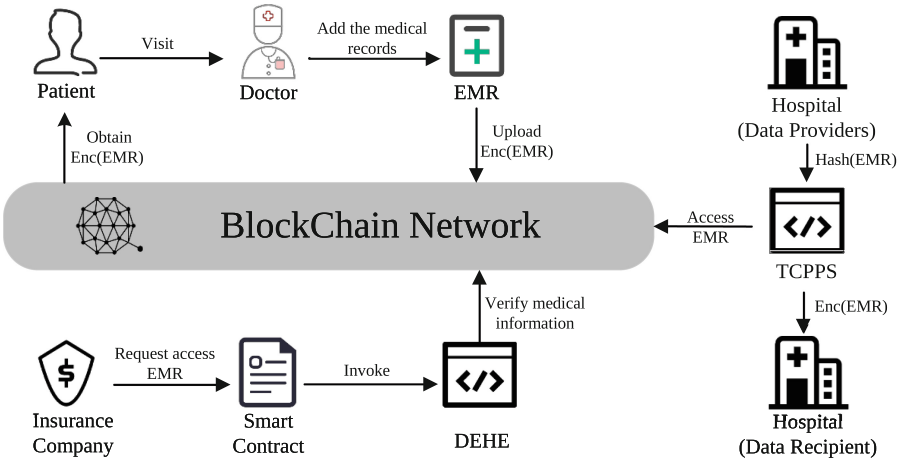


Fig. 1. The architecture of the PMI-Chain.

The PMI-Chain framework provides secure EMR management through a sequence of well-defined steps. Medical institutions generate and upload EMR to the blockchain for secure storage, based on patient authorization. Upon completing a diagnosis, physicians trigger a smart contract, which executes specific encryption processes, subsequently uploading the ciphertext to the blockchain. Patients then decrypt to access their updated medical records. When sharing information between medical institutions, hash medical data are exchanged via the blockchain. Patients possess exclusive Token tokens and ECC key pairs. The recipient retrieves the data using the Token and verifies it through a smart contract. Finally, patients use their private key to decrypt the data, ensuring secure retrieval. Insurance companies retrieve the EMR ciphertext via hash indices provided by medical institutions and validate these using claim verification smart contracts. Upon successful validation, a reimbursement calculation smart contract computes the claim amount. With this system, insurance claim processing

can be accomplished through a homomorphic encryption system without accessing the plaintext of the medical records, thereby ensuring both operational efficiency and patient privacy.

3.2 Token-Based Access Control for Blockchain Smart Contracts

In the core research of secure medical data sharing, we have constructed a token-based blockchain smart contract access control privacy protection scheme (TBSC-ACPPS). This scheme integrates the concept of tokens from the CapBAC model, and uses smart contracts to manage EMR retrieval as well as token generation and distribution. Its aim is to enhance platform efficiency, reduce human errors, thereby providing an efficient strategy for medical data sharing and privacy protection during referral processes.

We accomplish secure access control through carefully designed smart contracts and tokens, and employ Elliptic Curve Cryptography to ensure the security of EMR data during transmission. We will illustrate this part of the model through the detailed process of inter-hospital EMR sharing. The specific operations are as shown in Fig. 2.

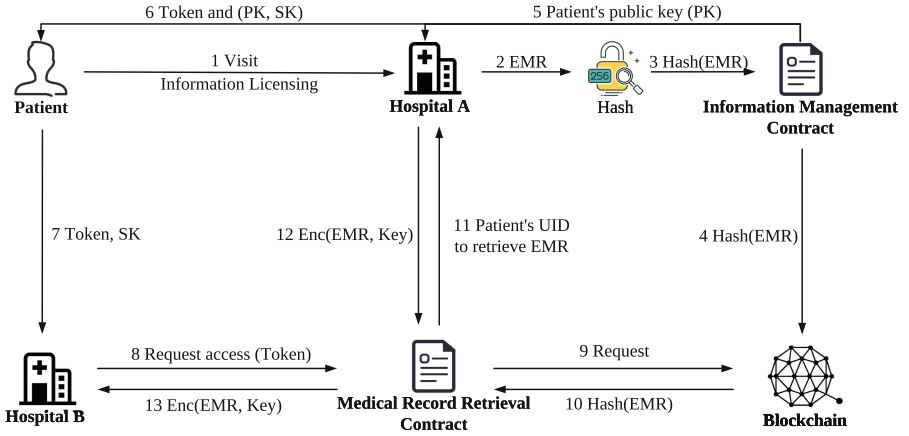


Fig. 2. Token-based access control for blockchain smart contracts.

Upon the conclusion of a patient's visit, the attending physician generates an EMR. This EMR undergoes a hash computation and is securely uploaded to the blockchain, with the smart contract producing a corresponding blockchain hash index. Simultaneously, a unique, patient-specific token and a cryptographic key pair are generated, with the public key sent directly to the initial hospital. In scenarios where the patient is referred to a different hospital, the incoming physician can retrieve the patient's EMR directly from the blockchain using this unique token. The smart contract verifies the patient's identity, accesses the blockchain-stored EMR hash with the token, and securely encrypts the EMR

before transmission. If the EMR hashes align, the smart contract dispatches both the EMR cipher and key cipher to the new physician. The patient, with the private key, decrypts the data to access the EMR in plain text. A pivotal feature of our TBSC-ACPPS is the robust token-based access control module. This module, encompassing token generation, transfer, verification, and updating, significantly enhances the system's security, optimizes resource utilization, and assures the trustworthy sharing of EMRs, alongside comprehensive patient privacy protection.

Smart Contract Design: There are three smart contracts in TBSC-ACPPS, which are information management contract, medical record retrieval contract and token update contract.

A. Information Management Contract. The patient's EMR need to be stored as evidence in the blockchain. To speed up the process of uploading information and optimize storage space, the TBSC-ACPPS only uploads the SHA-256 hash value of the medical record information ($patient\_EMRHash$) to the blockchain for storage.

$$patient\_EMRHash = SHA - 256(patient\_EMR) \tag{1}$$

After the contract uploads $patient\_EMRHash$ to the blockchain network and obtains the hash index, it generates an ECC key pair (PK, SK) and a unique Token for the patient, which is used for retrieving medical records.

The ECC key pair is generated by selecting a point G on the elliptic curve E as the generator and assuming the order of G to be n, where n is a prime number. Finally, a private key $k(k < n)$ is chosen to generate the public key $Q = kG$. The Token is the patient's exclusive Token, which includes the patient's unique identity ($patient\_UId$), the on-chain hash value ($blockChain\_hash$), the hospital's blockchain address ($hos\_address$), and a timestamp ($time$), as demonstrated below:

$$Token = \{patient\_UId,\ blockChain\_hash,\ hos\_address,\ time\} \tag{2}$$

In the end, the contract sends the Token and the key pair (PK, SK) to the patient, and also sends the public key PK to the hospital. This allows for the encrypted transmission of subsequent medical records, ensuring the privacy of the patient during the transmission process.

B. Medical Record Retrieval Contract. When a patient seeks treatment in another hospital, there is a need to retrieve the medical records from the previous hospital. The patient gives the Token to the doctor who uses it through the smart contract to access the patient's medical history. Initially, the contract verifies the operator's permissions and the Token. If successful, the contract retrieves the medical record hash ($blockChain\_EMRHash$) via the on-chain hash value index ($blockChain\_hash$) in the Token and sends the patient identity marker ($patient\_UId$) from the Token to the hospital using the hospital blockchain

address ($hos\_address$). The hospital then locates the original patient medical record via $patient\_UId$, calculates the SHA-256 hash of the record, encrypts the original record using symmetric encryption, encrypts the symmetric key (Key) using the patient's public key (PK), and sends the medical record's hash value ($hos\_EMRHash$), the encrypted medical record ($Enc(EMR)$), and the encrypted key ($Enc(Key)$) to the smart contract. Lastly, the contract compares the $blockChain\_EMRHash$ and $hos\_EMRHash$ from the blockchain. If they match, it proves that the medical record hasn't been tampered with, and the doctor receives the ($Enc(EMR)$) and ($Enc(Key)$) for the patient to decrypt using their private key (SK).

In the smart contract for retrieving medical records, the original ($patient\_EMR$) is encrypted first using symmetric encryption to get $Enc(EMR)$:

$$Enc(EMR) = E(Key, patient\_EMR) \tag{3}$$

And the key is encrypted as follows:

$$\begin{gathered} Enc(Key) = (Cipher1, Cipher2) \\ Cipher1 = Key + rQ \\ Cipher2 = rG \end{gathered} \tag{4}$$

where r is a random number ($r < n, n$ is the order of G), Q is the patient's public key, and $Cipher1$ and $Cipher2$ are two ciphertexts. Subsequently, the patient uses their ECC private key (SK) to decrypt, obtaining the $patient\_EMR$:

$$Cipher1 - k \times Cipher2 = Key + rQ - krG = Key + rkG - krG = Key \tag{5}$$

$$patient\_EMR = D(Key, Enc(EMR)) \tag{6}$$

where k is the patient's private key (SK).

C. Token Update Contract. When a patient's Token expires, the patient can request the Token Update Contract to refresh their Token. Initially, the smart contract verifies whether the patient possesses the requisite authority to request the token, then proceeds to update the token, which includes refreshing the timestamp.

3.3 Privacy Protection via Dual ElGamal Encryption in Blockchain Smart Contracts

In insurance claim scenarios, fraudulent medical claims often occur due to the difficulty in verifying the authenticity of reimbursement data provided by patients. Furthermore, processing medical insurance claims requires accessing patients' reimbursement data for proof of claim, which risks leaking the patient's private information. To address these issues, we propose a dual ElGamal homomorphic encryption privacy protection scheme based on blockchain smart contracts (DEHE-BSC). Leveraging the tamper-proof and traceable nature of the

blockchain provides secure evidence storage for reimbursement data, while smart contracts enable data to be recorded on-chain and facilitate claim data verification. The improved ElGamal homomorphic encryption technology is adopted to carry out medical insurance claims in ciphertext form, thereby safeguarding patient privacy.

We will describe a specific claims process to provide a detailed explanation of security claims. The specific steps are shown in Fig. 3.

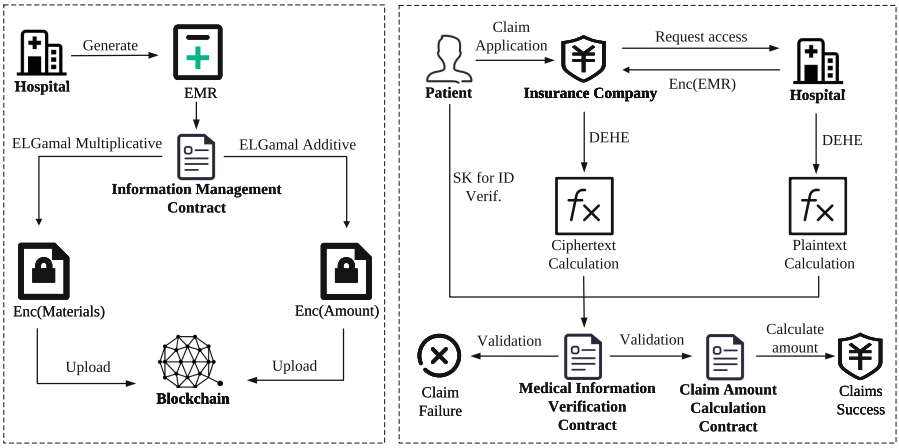


Fig. 3. Dual ElGamal homomorphic encryption privacy protection scheme based on blockchain smart contracts.

After the medical visit, the hospital generates the patient's EMR and activates the on-chain smart contract. This contract splits the EMR data and uses ElGamal homomorphic encryption to encrypt the minimal dataset required for insurance claims, along with the claim amount. These two encrypted pieces of data are then uploaded to the blockchain for secure storage. When a patient files a claim, the insurance company extracts the encrypted EMR data from the blockchain using the ciphertext hash value and executes a specific computation. The hospital performs the same operation on the corresponding plain-text EMR in its database. Both parties send their results to the claim verification smart contract, and the patient provides the decryption key. The contract automatically decrypts the insurance company's encrypted result and compares it with the hospital's plain-text result. If the results match, the claims smart contract retrieves the encrypted claim amount from the blockchain, performs the necessary computations, and finalizes the claim. If the results do not match, it indicates potential malicious tampering with the EMR data, and the insurance company may refuse the claim.

Smart Contract Design: The core content architecture of DEHE-BSC, which mainly includes the attestation of medical information verification, claim auxiliary verification, and claim amount calculation.

A. Medical Information Verification Contract. During the insurance claim process, insurance companies need to verify the authenticity of user data (*message*), as well as the accuracy of claim amounts. Considering the size and differing purposes of the data, to improve operational efficiency, the Medical Information Verification Contract splits the message into reimbursement data (*messageData*) and treatment cost (*messageCost*), each encrypted using different homomorphic encryption techniques.

For messageData, the ElGamal multiplication homomorphic technique, which offers faster encryption and decryption speeds, is used for encryption. This enables non-plaintext verification of the reimbursement data. The key pair (PK, SK) is generated as follows: p is a randomly selected large prime number, g is a generator of p, and x, k are random numbers. The public key PK is y, and the private key SK is x.

$$y = g^x \bmod p \tag{7}$$

The encrypted reimbursement material data ciphertext ($encMessageData$) is as follows:

$$\begin{gathered} encMessageData = (c_1, c_2) \\ c_1 = g^k \bmod p \\ c_2 = messageData \times y^k \bmod p \end{gathered} \tag{8}$$

For the claim amount *messageCost*, an additive variant of the ElGamal addition homomorphic encryption algorithm is used for encryption. This allows for non-plaintext computation of claim amounts. The encrypted treatment cost ($encMessageCost$) is as follows:

$$\begin{gathered} encMessageCost = (c_1, c_2) \\ c_1 = g^k \bmod p \\ c_2 = a^{messageCost} \times y^k \bmod p \end{gathered} \tag{9}$$

where a is a constant chosen at random.

B. Claim Auxiliary Verification Contract. After a patient initiates an insurance claim with the insurance organization, the insurance company must apply to the medical institution for the hash index to access the encrypted medical record on the chain: $encMessageData_1, encMessageData_2, encMessageData_3$ ... $encMessageData_n$. The multiplication result of the ciphertext data of multiple entries for this patient, encMessageDatamult, is obtained for verification.

$$encMessageData_{mult} = encMessageData_1 \times \cdots \times encMessageData_n \tag{10}$$

The hospital performs the same calculation on the patient's corresponding plaintext data of materials, obtaining the multiplication result of all the plaintext data of the materials, $messageData_{mult}$.

$$messageData_{mult} = messageData_1 \times messageData_2 \cdots messageData_n \tag{11}$$

The Claim Auxiliary Verification Contract performs consistency verification on the encrypted calculation results provided by the insurance organization and the hospital using the *Key* provided by the patient. If the verification passes, it indicates that there has been no data tampering, and the claim proceeds; otherwise, the claim is terminated.

C. Claim Amount Calculation Contract. After the consistency verification by the Claim Auxiliary Verification Contract, it implies that there are no malicious activities such as data tampering. Then, the Claim Amount Calculation Contract performs homomorphic calculations on the ciphertext of the claim amount and decryption operations on the calculation results. First, it fetches several ciphertexts of the claim amounts for this patient from the blockchain: $encMessageCost_1$, $encMessageCost_2, encMessageCost_3 ... encMessageCost_n$. Then, the contract calculates the multiple ciphertext data of this patient, obtaining the summation result of all treatment cost ciphertexts, $encMessageCost_{add}$.

$$encMessageCost_{add} = encMessageCost_1 + \cdots + encMessageCost_n \tag{12}$$

Next, the treatment cost ciphertext result, $encMessageCost_{add}$, is decrypted.

$$\begin{gathered} encMessageCost_{add} = (c_1, c_2) \\ a^{messageCost_{add}} = \frac{c_2}{{c_1}^x} = \frac{a^{messageCost_{add}} y^k}{g^{kx}} = \frac{a^{messageCost_{add}} g^{xk}}{g^{xk}} \bmod p \\ cost = messageCost_{add} = \log_a a^{messageCost_{add}} \end{gathered} \tag{13}$$

The *cost* is the plaintext of the real treatment costs of the patient, which has been verified for security and claim amount.

DEHE: Within the DEHE-BSC framework, the data receiver processes ciphertext encrypted using Dual ElGamal Homomorphic Encryption (DEHE), performing computations and decryption verification on the results. DEHE is built upon the ElGamal public-key cryptosystem and carries out multiplicative homomorphic encryption on the minimum EMR data set, leveraging its efficient encryption and decryption to verify reimbursement data swiftly. In parallel, DEHE employs an additive variant of the ElGamal encryption to encrypt the reimbursement amount, using its homomorphic properties for claims calculation. The security of DEHE stems from the complexity of the discrete logarithm problem in finite fields, which ensures the safety of the ciphertext. The specific process of the DEHE algorithm is illustrated in Algorithm 1.

The DEHE algorithm encompasses functions for key generation, encryption, decryption, and homomorphic computation. Its inputs can be plaintext, ciphertext, or computational data, and its outputs are key pairs, ciphertext, plaintext, or computational results. During key generation, the ElGamal algorithm generates a key pair for the patient. The encryption process divides plaintext data into reimbursement data and treatment costs, which are then encrypted using ElGamal multiplicative homomorphism and its additive variant, respectively. The decryption process utilizes the patient's private key to decrypt the ciphertext.

Algorithm 1: DEHE Algorithm

Input : *plainText* or *cipherText* or *message*
Output: *Key* or *encPlainText* or *decCipherText* or *calculationResult*

Function *genKey()*:
    `// Key Generation Function.`
    $SK \leftarrow x$;
    $PK \leftarrow g^x \mod p$;
    $Key \leftarrow SK + PK$;
    return *Key* // Return the key pair;
Function *encryption(plainText)*:
    `// Dual ElGamal Homomorphic Encryption Function.`
    $encPlainTextData \leftarrow Enc(plainText.data)$;
    $encPlainTextCost \leftarrow Enc(plainText.cost)$;
    $encPlainText \leftarrow encPlainTextData + \text{str}(encPlainTextCost)$;
    return *encPlainText* // Return the ciphertext;
Function *decryption(cipherText)*:
    `// Decryption Function.`
    $decCipherText \leftarrow Dec(cipherText, SK)$ // Decrypt using the private key;
    return *decCipherText* // Return the plaintext;
Function *homomorphicCalculation(message)*:
    `// Homomorphic Calculation Function.`
    if *message.type* == *data* **then**
        `// ElGamal multiplication homomorphic calculation.`
        calculationResult ← multElGamal(message) ;
    else if *message.type* == *cost* **then**
        `// ElGamal addition homomorphic calculation.`
        calculationResult ← addElGamal(message) ;
    end
    return *calculationResult* ;

The homomorphic computation function performs corresponding computations based on the data type.

4 Analysis

4.1 Correctness Analysis

Theorem 1. *Definition of Correctness Concern: Upon the insurance institution transmitting the computational outcome* $Enc(M_1)$ *to the smart contract, the contract is capable of invoking the patient's decryption key 'Key' to decipher it. The decrypted result is then juxtaposed with the unencrypted computational outcome* M_2 *from the hospital's end to validate the congruence of both datasets* M_1 *and* M_2*. This juxtaposition ascertains whether the data has undergone any unauthorized modifications.*

Proof. Let us presume the patient's unencrypted data to be m_1, m_2. Consequently, the ciphertexts calculated by the insurance institution will be $Enc(m_1)$ and $Enc(m_2)$. Given that ElGamal exhibits multiplicative homomorphism, it would employ the rules of multiplication, thereby yielding:

$$\mathrm{Enc}\,(M_1) = \mathrm{Enc}\,(m_1) \times \mathrm{Enc}\,(m_2) \tag{14}$$

Given the circumstances,

$$\begin{aligned} \mathrm{Enc}\,(M_1) &= \mathrm{Enc}\,(m_1) \times \mathrm{Enc}\,(m_2) \\ &= \left(g^{k_1}, m_1 y^{k_1}\right) \times \left(g^{k_2}, m_2 y^{k_2}\right) \\ &= \left(g^{k_1+k_2}, m_1 m_2 y^{k_1+k_2}\right) \\ &= \mathrm{Enc}\,(m_1 m_2) \end{aligned} \tag{15}$$

Therefore, post-decryption of $Enc(M_1)$ using the decryption Key, we obtain M1, the value of which is equivalent to $m_1 m_2$. Since M_2 employs identical multiplication operations, the resultant value of M_2 is also $m_1 m_2$. Thus, when M_1 equals M_2, it can be inferred that the patient data has remained untampered.

4.2 Security Analysis

Theorem 2. *Definition of Security Concern: An insurer should not be able to infer any information about* $m_1, m_2, ...m_n$ *using* $Enc(m_1), Enc(m_2), ...Enc(m_n)$.

Proof. The well-established ElGamal encryption scheme's security relies on the inherent difficulty of the Discrete Logarithm Problem in finite fields, and the insurer can only access the patient's ciphertext.

Suppose the insurer obtains encrypted data $Enc(m_1), Enc(m_2), ...Enc(m_n)$ and is privy to the public key y, a large prime number p, and the generator g. Given that the patient's private key x is kept secure, the insurer would have to reverse compute the value of x from the equation below to obtain the plaintext $m_1, m_2, ...m_n$:

$$y = g^x \bmod p \tag{16}$$

This computation belongs to the Discrete Logarithm Problem, a notoriously hard problem that consists of determining the value of x when a large prime number p, its corresponding generator g, and a value y are given. The difficulty of computing x increases exponentially with the size of p.

Therefore, when the value of p is sufficiently large, it can be asserted that this scheme is secure, implying that the insurer cannot decipher the patient's plaintext m using the ciphertext $Enc(m)$.

4.3 Privacy Analysis

Patient medical records are not directly stored on the blockchain. Instead, we use a double ElGamal encryption algorithm to encrypt these records and store them

within medical institutions, while the blockchain only saves the associated hash values. This approach significantly reduces the possibility of patient medical data leakage. Even if the blockchain is maliciously attacked, patient privacy cannot be compromised. In addition, we proposed a token-based access control strategy that achieves fine-grained access control and deep privacy protection. Only users with the corresponding tokens can retrieve the encrypted medical records through smart contracts and finally decrypt the patient's medical records using the user's private key.

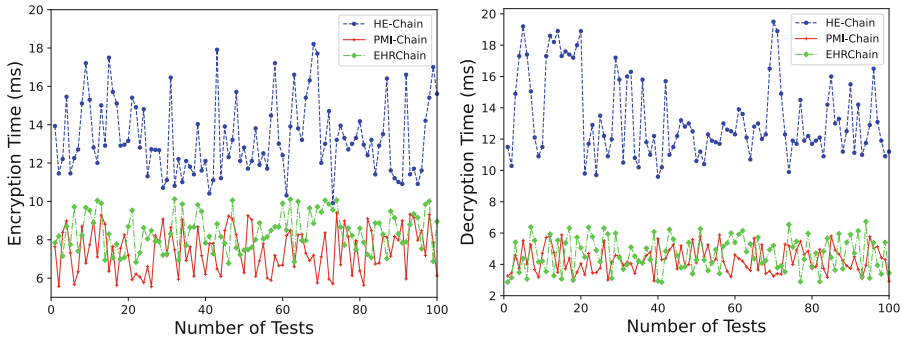
In the insurance claims process, insurance companies can only receive the ciphertext of patient medical information, and under a secure encryption scheme, this ciphertext is undecipherable. Through the consistency verification mechanism of DEHE, insurance companies can only obtain the calculation result and cannot access specific patient medical records, thereby ensuring the privacy of patient-sensitive data during interactions with third parties.

4.4 Efficiency Analysis and Comparison

We set up a local Ethereum blockchain network using the Geth client and constructed a set of smart contracts deployed on the channel. The DPPChain platform was implemented on a PC equipped with an Intel Core CPU i7-10875H and a GeForce RTX 2070 GPU. For privacy protection, we implemented Dual ElGamal Homomorphic Encryption using Java. For interaction between the system and blockchain data, we used Ethereum's official web3j to handle the interchange process between Ethereum and the application layer. Next, we will delve into the each part of the PMI-Chain system and discuss its efficiency.

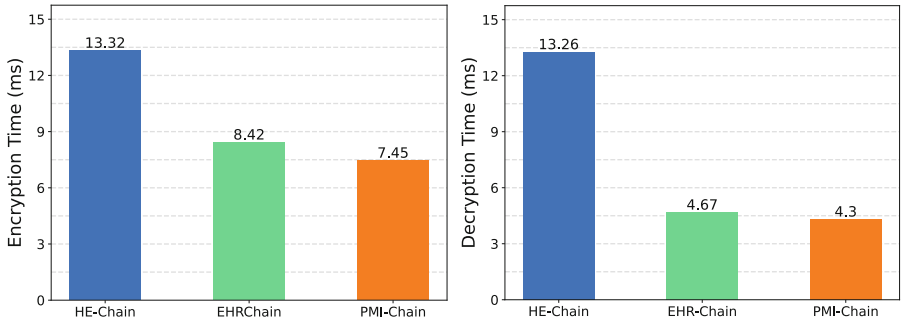
Efficiency of DEHE-BSC: In this section, we tested the encryption and decryption performance of the DEHE-BSC scheme executed by PMI-Chain during the insurance claim process, and compared it with the methods proposed in two related papers. Wenyu et al. [15] first introduced a privacy protection mechanism for insurance claims into the medical information platform, implementing the claims process using basic homomorphic encryption, which we refer to as HE-Chain. The EHRChain system proposed in [2] is a medical information processing platform with superior overall performance. It uses an improved homomorphic encryption technique, effectively reducing the computational load during encryption and decryption processes, thereby enhancing the model's efficiency.

Performance Testing of Encryption. As shown in Fig. 4a and Fig. 5a, compared to HE-Chain and EHRChain, the encryption strategy adopted by PMI-Chain increases the encryption speed by 44% and 12% respectively, and its performance is more stable. This is mainly because PMI-Chain uses an improved ElGamal encryption algorithm, whose encryption process mainly involves exponential and modular exponentiation operations with fewer number theory operations involved. These operations are relatively fast in computation, and they can be accelerated by using efficient algorithms such as fast power algorithm. On the other hand, the encryption process of the Paillier algorithm used in HE-Chain



(a) Encryption Time (b) Decryption Time

Fig. 4. Comparison of the time to perform 100 encryption and decryption tests for HE-Chain, PMI-Chain and EHRChain.

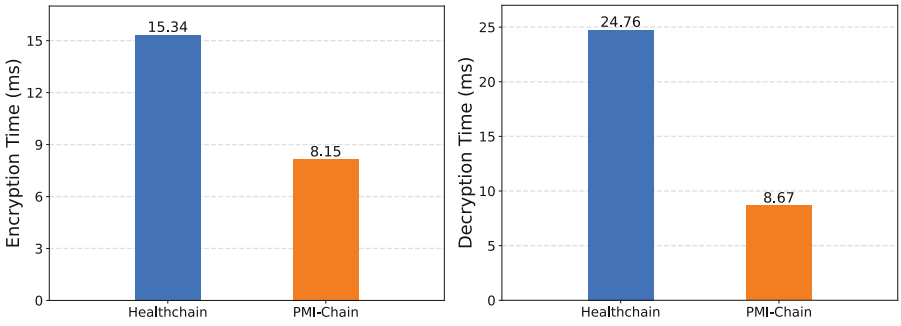


(a) Encryption Time (b) Decryption Time

Fig. 5. Compare the average time of 100 encryption and decryption tests performed by HE-Chain, PMI-Chain and EHRChain.

involves multiplication and modulus operations of large numbers, as well as some complex number theory operations. These operations are relatively slower, and as the length of the key increases, the computation time will increase as well. Although EHRChain has improved the basic Paillier algorithm, its key generation process still involves more complex number theory calculations, including the generation of large primes and some complex number theory operations, which could make key generation more time-consuming. On the contrary, in the PMI-Chain, the key generation process usually only involves steps such as selecting a prime number and generating random numbers, making it more efficient.

Performance Testing of Decryption. As shown in Fig. 4b and Fig. 5b, in terms of decryption time, the performance of PMI-Chain has increased by 67% and 8% respectively compared to HE-Chain and EHRChain. Similar to the encryption process, this is because the operations such as exponentiation and modular exponentiation involved in the decryption process of DEHE can be accelerated



(a) Encryption Time (b) Decryption Time

Fig. 6. Compare the average time of 100 encryption and decryption tests performed by Healthchain and PMI-Chain.

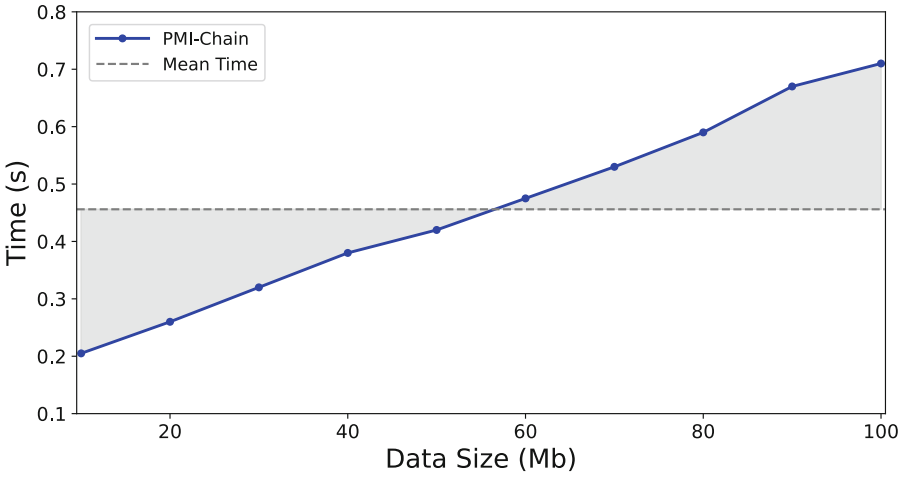


Fig. 7. Time of token generation under different data sizes

by efficient algorithms. In contrast, the decryption process based on the Paillier algorithm involves multiplication and modulus operations of large numbers, and its decryption operations are more complex, requiring more number theory computations, thereby resulting in a relatively longer decryption time.

Efficiency of TBSC-ACPPS: In this section, we test the encryption, decryption, and token generation performance of the TBSC-ACPPS scheme implemented by PMI-Chain during the process of medical institutions data sharing. We compare it with Healthchain [16]. Healthchain has designed a system that ensures privacy protection of medical data, thereby realizing reliable data sharing between medical institutions.

Performance Testing of Encryption: We conducted 100 encryption tests on the aforementioned schemes, taking the average of the results as the final data, as shown in Fig. 6a. The results show that the encryption processing time for Healthchain is 15.34 ms, while our PMI-Chain only takes 8.15 ms. It is evident that PMI-Chain offers the best encryption performance with the least time consumption.

Performance Testing of Decryption: As shown in Fig. 6b, the decryption test results indicate that the decryption processing time for Healthchain is 24.76 ms, and our PMI-Chain only takes 8.67 ms. This once again proves that PMI-Chain outperforms the other schemes in terms of decryption performance. This can be attributed to the lower computational complexity of PMI-Chain compared to Healthchain. Healthchain's medical record encryption scheme is more complex than PMI-Chain, resulting in a longer encryption time. It employs more complex number-theoretical computations compared to the point operations on the elliptic curve used by PMI-Chain, leading to a longer decryption time.

Time Consumption for Token Generation: Besides encryption and decryption activities, the speed at which tokens are generated plays a vital role in determining the overall system performance. Within the TBSC-ACPPS system, each patient token principally includes elements such as the User ID (UId), an on-chain hash value ($hash$), the address of the hospital's blockchain ($address$), and a timestamp ($time$).

We conducted a series of tests on the speed of token generation using medical record samples of varying sizes, ranging from 10 MB to 100 MB. As depicted in Fig. 7, the TBSC-ACPPS system effectively confines the token generation time to under one second. Even when the size of the medical record data in the test is 100MB, the time required to generate a token remains impressively low, at just 0.716 s.

5 Conclusions

In conclusion, this paper has presented the PMI-Chain, an innovative system that employs token-based smart contracts and dual ElGamal homomorphic encryption to establish a secure, confidential, and fine-grained access control-enabled data sharing platform for patients, medical institutions, and insurance companies. The platform addresses significant challenges inherent in traditional EMR systems, such as the risk of data breaches, limited patient autonomy over medical records, fragmented and inefficient allocation of medical resources, and potential violations of patient privacy. Our analysis demonstrates that the PMI-Chain provides a significant performance improvement over existing methods, with lower encryption and decryption time overhead and more stable performance.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (61962042) and Science and Technology Program of Inner Mongolia Autonomous Region (2020GG0188), and Natural Science Foundation of Inner Mongolia (2022MS06020), and the Central Government Guides Local Science and Technology

Development Fund (2022ZY0064), and the University Youth Science and Technology Talent Development Project (Innovation Group Development Plan) of Inner Mongolia A. R. of China (Grant No. NMGIRT2318).

References

1. Hillestad, R., et al.: Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. Health Aff. **24**(5), 1103–1117 (2005)
2. Li, F., Liu, K., Zhang, L., Huang, S., Wu, Q.: EHRchain: a blockchain-based EHR system using attribute-based and homomorphic cryptosystem. IEEE Trans. Serv. Comput. **15**(5), 2755–2765 (2021)
3. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: MedRec: using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD), pp. 25–30. IEEE (2016)
4. Zhu, W.: Research on theoretica framework on excessive medica treatment and hierarchica medical system based on the perspective of supply side. Chinese Health Econ. **37**(3), 8–10 (2018)
5. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized business review (2008)
6. Stafford, T.F., Treiblmaier, H.: Characteristics of a blockchain ecosystem for secure and sharable electronic medical records. IEEE Trans. Eng. Manage. **67**(4), 1340–1362 (2020)
7. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **24**(1), 131–143 (2012)
8. Xhafa, F., Feng, J., Zhang, Y., Chen, X., Li, J.: Privacy-aware attribute-based PHR sharing with user accountability in cloud computing. J. Supercomput. **71**, 1607–1619 (2015)
9. Cao, S., Zhang, G., Liu, P., Zhang, X., Neri, F.: Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. Inf. Sci. **485**, 427–440 (2019)
10. Saini, A., Zhu, Q., Singh, N., Xiang, Y., Gao, L., Zhang, Y.: A smart-contract-based access control framework for cloud smart healthcare system. IEEE Internet Things J. **8**(7), 5914–5925 (2020)
11. Benil, T., Jasper, J.: Cloud based security on outsourcing using blockchain in e-health systems. Comput. Netw. **178**, 107344 (2020)
12. Ding, Y., Sato, H.: Derepo: a distributed privacy-preserving data repository with decentralized access control for smart health. In: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp. 29–35. IEEE (2020)
13. Zhang, L., et al.: BDSS: blockchain-based data sharing scheme with fine-grained access control and permission revocation in medical environment. KSII Trans. Internet Inf. Syst. **16**(5), 1634–1652 (2022)
14. Dewangan, N.K., Chandrakar, P.: Patient-centric token-based healthcare blockchain implementation using secure internet of medical things. IEEE Trans. Comput. Soc. Syst. **10**, 3109–3119 (2022)
15. Xu, W., Wu, L., Yan, Y.: Privacy-preserving scheme of electronic health records based on blockchain and homomorphic encryption. J. Comput. Res. Dev. **55**(10), 2233–2243 (2018)
16. Wang, B., Li, Z.: Healthchain: a privacy protection system for medical data based on blockchain. Future Internet **13**(10), 247 (2021)