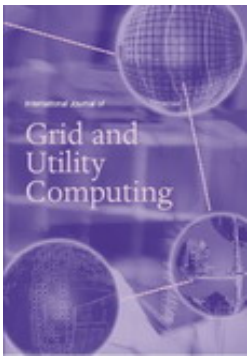


International Journal of Grid and Utility Computing

ISSN online: 1741-8488 - ISSN print: 1741-847X
https://www.inderscience.com/ijguc

Performance evaluation using throughput and latency of a blockchain-enabled patient centric secure and privacy preserve EHR based on IPFS

Vishal Sharma, Niranjan Lal, Anand Sharma

DOI: 10.1504/IJGUC.2023.10059539

Article History:

Received:	07 April 2023
Last revised:	22 June 2023
Accepted:	09 July 2023
Published online:	19 February 2024

Performance evaluation using throughput and latency of a blockchain-enabled patient centric secure and privacy preserve EHR based on IPFS

Vishal Sharma

CSE Department (SET),
Mody University of Science and Technology,
Lakshmangarh, Sikar, Rajasthan, India
Email: er.vishu1983@gmail.com

Niranjan Lal*

Department of Computer Science and Engineering,
SRM Institute of Science and Technology (Delhi-NCR Campus),
Modinagar, Ghaziabad, Uttar Pradesh, India
Email: niranjan_verma51@yahoo.com
*Corresponding author

Anand Sharma

CSE Department (SET),
Mody University of Science and Technology,
Lakshmangarh, Sikar, Rajasthan, India
Email: anand_glee@yahoo.co.in

Abstract: Every nation needs a better healthcare system and services for general people for digital medical records, which are available on a large scale. However, patients' health data is too sensitive to share and unsecured to store on centralised storage. However, it is required to ensure security and privacy with better storage and retrieval methods for PEHR (Patient Electronic Health Record). Blockchain allows for the secure and effective exchange of PEHR in a decentralised, tamper-proof manner and traceable distributed ledger that stores using Hyperledger Fabric (HLF) framework in encrypted form on the InterPlanetary File System (IPFS). The hyperledger caliper benchmark measures the blockchain network's performance concerning transaction throughput and latency. This paper discusses the performance evaluation of a Blockchain-Enabled Patient Centric Secure (BEPCS) and privacy preserved electronic health record on IPFS. It proposes a strategy that may increase throughput by 5–10% and decrease latency by 5–10% with better security and privacy.

Keywords: medical data security; patient electronic healthcare records; consortium blockchain; inter planetary file system; medical data privacy preservation; chaincode; proxy re-encryption; patient-directed healthcare system.

Reference to this paper should be made as follows: Sharma, V., Lal, N. and Sharma, A. (2024) 'Performance evaluation using throughput and latency of a blockchain-enabled patient centric secure and privacy preserve EHR based on IPFS', *Int. J. Grid and Utility Computing*, Vol. 15, No. 1, pp.16–30.

Biographical notes: Vishal Sharma is currently doing research in the Department of Computer Science and Engineering, Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan, India. His current research interests include rational secret sharing and distributed computing.

Niranjan Lal is currently an Associate Professor & Head of Computer Science and Engineering (DSBS) at SRMIST, Delhi NCR Campus, Ghaziabad, Uttar Pradesh, India. He has over 15 years of extensive experience in the IT industry and academics. His research areas include network security, wireless sensor networks, cloud computing, dataspaces, android application development, information retrieval, machine learning, IoT and blockchain.

Anand Sharma has been an Assistant Professor in the Department of Computer Science and Engineering, Mody University of Science and Technology, Lakshmangarh, Sikar, Rajasthan, India, for the last 12 years. He has more than 14 years of experience in teaching and research. He has pioneered research in Information Security, IoT, ML, WBAN and HCI.

1 Introduction

The increasing amount of healthcare data generated daily presents a significant challenge for healthcare providers to manage, store and share this data in a secure and privacy-preserving manner. Traditional electronic health record (EHR) systems have several limitations, including a lack of interoperability, data silos and privacy concerns. Blockchain Technology (BT) has emerged as a promising solution for secure and decentralised data management, but there are still challenges related to scalability, performance and patient-centricity.

The paper addresses these challenges by proposing a novel blockchain-based EHR system that prioritises patient-centricity, security and privacy preservation. The system uses InterPlanetary File System (IPFS) to store patient health data, which provides a distributed and resilient storage solution that ensures data availability and accessibility.

The BT ensures the integrity and immutability of patient health data and provides a transparent and secure platform for sharing and accessing this data. The proposed system also gives patients complete control over their health data, including granting and revoking access to healthcare providers and researchers.

This paper focuses on two key performance metrics, throughput and latency, to evaluate the system's performance. Throughput measures the number of transactions the system can process per second, while latency measures the time it takes for a transaction to be confirmed and added to the blockchain. The experiments conducted by the authors demonstrate that the proposed system has high throughput and low latency, which indicates its potential for use in real-world healthcare settings.

Technology improvements throughout the past century have led to an evolution in healthcare records management, storing, exchanging and analysing patient health records. Instead of physically noting a patient's diagnosis and course of treatment on paper, maintaining their health records is now done digitally. Digital medical records, also known as EHRs, are projected to be shared regularly across healthcare providers, including physicians, hospitals, chemists, insurance agencies, patients and healthcare researchers. Despite their reliability and convenience, traditional patient digital medical record systems pose several risks related to the security and privacy of medical data. The digital medical record is the most sensitive data-collecting method because it contains much private information about patients and diagnoses (Sharma et al., 2022). However, as the internet and digital healthcare systems have improved, EHR (Verma and Sharma, 2022) data has become more vulnerable to hacking. The lives and property of patients are seriously at risk in the traditional healthcare records administration system, where each institution keeps its database of patients' medical records. Alternatively, a centralised cloud server can pose considerable data privacy leakage and misuse concerns. As a result, medical data privacy and security issue during the data exchange process is of great concern. Additionally, it is challenging to connect numerous hospitals so that they may exchange these records and work together effortlessly for the common good. The findings show that there is an increasing amount of data breaches that compromise private healthcare information.

Healthcare data breaches occur due to unauthorised access that steals sensitive information about healthcare organisations and their patient records. The information typically targeted includes personal and medical records, insurance information and financial data. Some common causes of healthcare data breaches include: 1) Employee error: This can include accidental disclosure of information, such as sending an email to the wrong person or losing a laptop or mobile device. 2) Cyberattacks: These can come as hacking, malware or phishing scams. Cybercriminals can use these methods to access healthcare systems and steal sensitive information. 3) Insider threats can come from current or former employees who intentionally steal or misuse data. 4) Third-party breaches can occur when a vendor or contractor with access to healthcare data experiences a breach. 5) Physical theft can include stealing paper records, laptops or mobile devices containing sensitive information.

The Health Insurance Portability and Accountability Act (HIPPA), established in 1996, is a law that protects these medical records. However, there were almost three times as many data breaches (https://www.hipaajournal.com, 2023) between 2017 and 2023. Table 1 shows the healthcare data breaches and their causes between 2017 and 2023.

Table 1 Healthcare data breaches and causes between 2017 and 2023

Year	*Number of breaches*	*Causes of breaches*
2017	358	#1, #2, #3, #4
2018	369	#1, #2, #3, #4
2019	512	#1, #2, #3, #4
2020	663	#1, #2, #3, #4
2021	715	#1, #2, #3, #4
2022	707	#1, #2, #3, #4, #5
2023	40	#1, #2, #3, #4, #5

Note: Healthcare Data breaches of 500 or more records, as of March 2023; HIPPA.

Breaches Causes – Hacking/IT-#1, Employee error-#2, Theft-#3, Unauthorised access-#4, Other-#5.

Figure 1 shows EHR data breaches of 500 or more records and depicts that increment in the last two years. So, it is essential to take proactive steps to prevent these breaches in the healthcare organisation with robust cybersecurity measures implementations, training for employees regularly data handling procedures, conducting regular risk assessments and developing a plan to handle the incidents.

As per the HIPPA Report (Alder, 2023) and (National Institutes of Health (NIH), 2023), 30% of breaches occur in hospitals, $100 million loss and 36% of medical complications due to ransomware attacks, 51% of healthcare industries reported increases in data breaches. 61% of breaches were reported due to employees' mistakes, and 337 breaches were reported in the first quarter of 2022 only and affected approximately two crore individuals in the USA. As a result, it is imperative to develop a secure PEHR sharing system that can address the problems with the currently available centralised PEHR sharing systems while preserving the reliability and integrity of a PEHR and protecting patient privacy.

In order to address the issues with the current PEHR sharing system, decentralisation of the system has been proposed. The blockchain is a distributed ledger with decentralisation, reliability and tamper-proof. Therefore, blockchain offers a potential alternative to centralised cloud storage. Blockchain platforms can work in private or public environments, and patient health information should not be made generally accessible due to its sensitivity (Xi et al., 2022). This paper proposed a blockchain-enabled, patient-centric, privacy-preserving and secure digital healthcare system based on HLF and IPFS for medical data sharing. The HLF is employed to construct a permissioned consortium BT for digital healthcare. Each file in IPFS, a Peer-to-Peer (P2P) distributed file system, is given a hash value so that users can use it to locate the related file (Nyaletey et al., 2019). Compared to the public blockchain, the Consortium Blockchain (CB) offers the advantage of secure storage.

Additionally, because the CB does not require network-wide confirmation, it has higher throughput and efficiency. Also, a mechanism for identity authentication was implemented using CB (Sharma and Lal, 2020). The CB network can provide more vital privacy protection because only authorised users can access it. Additionally, the CB network upholds smart contracts, and since running smart contracts does not cost any fee, user access control measures are implemented using smart contracts. The member hospitals jointly maintain each node in the CB, using the same data and access policy, enabling efficient PEHR exchange. More oversized items, such as videos and images, currently need to be stored directly in the blockchain and PEHR frequently contains these kinds of large files. Traditional systems will have issues with storage and sharing if large files are stored on the local databases. Adopting cloud storage raises the possibility of data misuse and privacy leaks caused by third parties.

Figure 1 (A) Healthcare data breaches and its causes between 2017 and 2022 (B) Causes of data breaches in 2021 and 2022 (C) Data breaches in PEHR in 2021 and 2022 (see online version for colours)

Consequently, we have introduced the IPFS here. Users can use the hash code to look for the related file by giving each file in IPFS a unique hash code. In our proposed system, the CB stores metadata (Liu et al., 2020), while IPFS stores encrypted PEHR. The only thing that users can do with a file on the blockchain is obtain its metadata, which is kept on the blockchain ledger and only accessible by the owner of the file or other legitimate users. Furthermore, we employ proxy re-encryption technology, which addresses both security and efficiency, to encrypt PEHR to secure patient health records' security further. In our proposed system, a patient centred access control is accomplished using Common Policy – Attribute Based Access Control (CP-ABAC) (Zhang et al., 2022) technology and auditing concepts. This system lets patients and healthcare professionals choose who has the right to access PEHR in order to handle data securely and prevent data fraud. Therefore, only data requesters with patient consent and whose attributes comply with the access rules can decode the PEHR. The current implementation also supports urgent situations where the patient cannot react or provide consent for data sharing. In our empirical performance analysis, the primary benchmarking tool was Hyperledger Caliper. Average latency, throughput and success rate are all included in our analysis. The configuration with changing the workload (number of transactions) and end users (number of nodes) was the main focus of the analysis.

This paper highlights the potential of BT to address the challenges of secure and privacy-preserving healthcare data management. It proposes a novel patient-centric solution prioritising data security, transparency and accessibility. The experiments' results demonstrate the proposed system's performance and scalability and provide a promising foundation for future research.

The remainder of the paper is divided into the following sections: We examine the existing study on blockchain and PEHR in Section 2 of this article. We explain our suggested method in Section 3. We assess the system's performance in Section 4. In Section 5, we conclude the entire paper.

2 Related works

This part relates to the use of BT in exchanging medical records regarding security, access control, privacy protection, transaction throughput and latency.

Venkatesan et al. (2021) proposed a blockchain and IPFS-based digital medical record system. They have analysed the comparisons of latencies and overhead of various matrices. Chen et al. (2021) presented a consortium blockchain-enabled (Hyperledger Fabric) healthcare information exchange system with searchable keyword encryption, K-anonymity and attributes-based access control to achieve data security and privacy-preserving among multiple healthcare entities. By modelling different rates of medical data access and different numbers of healthcare facilities, they have looked at the computational costs involved with encryption techniques, the efficacy of the proposed chain codes and the scalability of the recommended system. Pradhan et al. (2022) proposed a multi-organisation, multi-host, the on-chain and an off-chain framework for storing a patient's medical data as well as multiple peer-based schemes for an HLF-enabled medical system that discourses the challenges of data privacy, data availability and security of healthcare data using Google Cloud Platform. They employed Hyperledger Calliper, tcpdump to generate realistic traffic over the network, orderer for RAFT and Kafka for their performance analysis. They also compared the orderer services provided by Kafka and RAFT, and discovered that RAFT was better suited for open, query and client-side transfer activities. A Quorum consortium blockchain deployment on the Tencent Cloud, together with smart contracts for the sharing of health data, has been recommended by Zhang et al. (2022). They have also contrasted the suggested model with the current methodology and demonstrated its effectiveness regarding medical data security. Using HF and IPFS for on-chain and off-chain storage of EHR and health data files, respectively, Li et al. (2021) presented a system for efficiently storing and disseminating Electronic Healthcare Records (EHR). To safeguard privacy and make effective EHR sharing possible, they have also used an access control method that incorporates chaincode and CA components. They have compared the currently used standard systems and evaluate the effectiveness of the suggested system. According to Jain and Jat (2021) permissioned blockchain has been established for the healthcare industry using HLF, and a blockchain-based smart contract for storing health information using Go language. They have set up efficient healthcare systems to make transactions with many end users easier and safer. Wu et al. (2021) suggested a blockchain-based way to safeguard medical system private information. Through simulated experiments, in terms of the efficiency of information transmission, storage and control over security, they evaluate this approach's effectiveness. In order to effectively safeguard users' personal information, the information in its internal storage has been encrypted using the Elliptic Curve Diffie-Hellman (ECDH) Key Exchange (Wu et al., 2021). The issue of securely sharing and storing EHRs has been addressed by Sun et al. (2020) by offering a framework based on blockchain and smart contract technologies. The encrypted electronic records employ an encrypted keyword index and attribute-based encryption and can only be unlocked by attributes that agree with the access policy. They have analysed three criteria to evaluate performance: smart contract costs, cryptographic algorithm time costs and scheme characteristics. Ismail et al. (2020) suggested BlockHR, a blockchain-based health records management system, allow users to upload and access healthcare data in real-time for patients and healthcare providers for improved prediction and diagnosis of diseases. By using regular data and their present health, they have developed a prediction tool that enables network participants and outside users to calculate their likelihood of getting a disease. They analysed the proposed system's performance in protecting users' privacy and security from the risks associated with the client-server paradigm. According to their analysis and comparison of

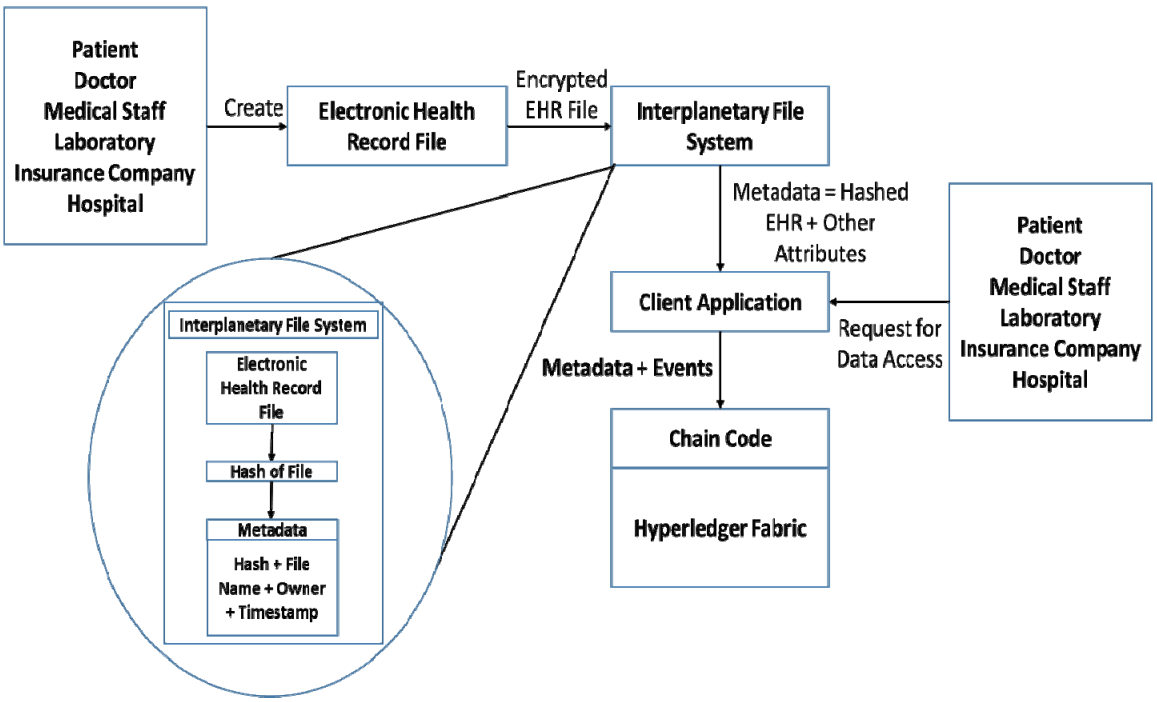
performance in terms of processing time for data reads and writes, the client-server approach takes 2.6 times a little less time than BlockHR to write medical file data. However, the read operation in the suggested framework is 20 times quicker than the client-server technique. Abunadi et al. (2019) proposed a patient centric blockchain enabled framework for securely share medical data between different users. They have simulated and analysed the proposed system against a centralised system for efficient health data protection. Blockchain-based patient medical records systems were proposed by Yazdinejad et al. (2020). They compared their research with two other approaches and looked at boosting network throughput while lowering overhead, speeding up reaction times, and using less energy (Sharma and Lal, 2022). Raising throughput of transactions between 3000 (tps) and 20,000 (tps) transactions per second, Gorenflo et al. (2020) presented a modified permissioned blockchain architecture called Hyperledger Fabric (HLF). They developed lightweight transactions, parallel validation and lightweight data structures for quick data access. They have determined that their work is superior to the seven earlier works. The HLF blockchain's throughput and latency measured by Herwanto et al. (2021). The infrastructure consists of 8 nodes and can handle up to 20,000 transactions per second. They examined 20,000 transactions and discovered the throughput and latency for a specific number of transactions using of HLF, v0.6 and v1.0, have evaluated their performance by Nasir et al. (2018). By adjusting the workload on each platform up to 10,000 transactions, they have looked at the two systems' throughput, latency and execution time performance. They could evaluate the two platforms' scalability by increasing each platform's node count to 20. Shuaib et al. (2022) suggested a permissioned blockchain-enabled system for exchanging healthcare data, and implemented on the Hyperledger Besu enterprise Ethereum blockchain. The proposed system makes use of the Interplanetary File System (IPFS) and the Istanbul Byzantine Fault Tolerant (IBFT) consensus method. They have assessed and contrasted the proposed framework's efficiency using a range of performance metrics, including transaction throughput, latency and failure rates. The studies decussated on varying network size and volume of transactions. The OmniPHR architectural concept, which incorporates distributed health records leveraging the openEHR interoperability standard and blockchain technology, was offered by Roehrs et al. (2019) as a prototype implementation and assessment. They evaluated the performance of incorporating medical records from several operational databases and the proposed prototype. They also took into account non-functional performance requirements like CPU utilisation, reaction time, disc usage, memory occupancy and access to networks in their evaluation criteria. In order to create a single view of health records, they have also investigated how to distribute and reintegrate the data. Nguyen et al. (2019) using a mobile cloud platform suggested a blockchain-based, decentralised IPFS architecture for sharing EHR system, and they developed a smart contract for authorisation. They have evaluated the effectiveness of the recommended method by putting up an Ethereum-based blockchain on Amazon's cloud. They have examined minimal network latency and safe data sharing. A patient control permissioned blockchain-based shared EHR system named MedBloc presented by Huang et al. (2022) as a way to integrate New Zealand's fragmented health IT ecosystem. They have suggested an encryption system and smart contract for access control to protect privacy and prevent illegal access. In order to compute the network's throughput and latency and to compare the study to already available solutions, the performance of the proposed system must be determined. With the use of the Hyperledger Calliper performance evaluating tool, Da Silva Costa et al. (2022) looked at their work and proposed a blockchain-based architectural design utilising HLF. They tested their system by expanding the network capacity from 3 to 13 peers, analysing average latency and throughput were calculated as critical criteria for their investigation, with the workload varying between 100 and 2500 concurrent record submissions. Mani et al. (2021) suggested an IPFS integrated framework and a patient-centric block chain for the off-chain and on-chain storage of healthcare data, respectively. Utilising hyper ledger calliper benchmarks for transaction latency, utilisation of resources and transactions per second (tps), they assessed the quality of their work.

Wen (2023) proposed a DPBFT consensus algorithm for digital music data copyright protection. He has analysed the comparison of the proposed algorithm with the existing algorithm and found that the average throughput can reach 1249 transactions per second (tps). Andriamanalimanana et al. (2020) discussed the three variant injection protocols for enhancing the performance of the blockchain network by decreasing the latency.

3 Proposed model

The proposed PEHR system is described in this section. The entities that comprise the system's components are defined, along with each entity's function, in the proposed PEHR system section. Figure 2 displays the proposed PEHR system's system model.

Figure 2 Proposed blockchain and IPFS-based PEHR system model



3.1 Preliminari

3.1.1 Blockchain

The first peer-to-peer cryptocurrency system using blockchain (Walia et al., 2022) as its foundation was described in a paper by Nakamoto (2008). A *blockchain* database stores a time-stamped collection of immutable information called blocks connected by cryptography and controlled by a group of independent nodes with no central authority (Liu et al., 2020). The various users can securely exchange their personal and professional records with each other using this framework (Alkouz et al., 2021).

The essential elements of blockchain architecture can vary depending on the specific blockchain implementation but typically include the following:

- *Block*: Each block in the chain has a distinct cryptographic hash that connects it to the preceding block in the chain. It is a collection of transactions that are confirmed and added to the blockchain.
- *Chain*: A sequence of blocks connected in chronological order makes up the decentralised, distributed ledger known as the blockchain. An unbreakable chain of blocks results from the fact that each block in the chain has a hash of the one before it.
- *Transaction*: A transaction is an exchange of data or value between two parties recorded on the blockchain. Transactions can include data such as digital assets, identity information or contracts.
- *Node*: An entity that keeps a copy of the blockchain ledger and approves transactions is a participant in the blockchain network. There are two types of nodes: full nodes, which maintain a complete copy of the blockchain, and lite nodes, which maintain a partial copy of the blockchain.
- *Consensus protocol*: The method or set of guidelines that controls how nodes in the network come to agree on the blockchain's current state is known as the consensus protocol. Consensus is necessary to ensure that all nodes have a duplicate copy of the blockchain and that no single node can manipulate the blockchain.

3.1.2 Types of blockchain

Permissionless (Public) blockchain: In this ecosystem, a node can join or leave the network at any time and carry out transactions. An enormous number of nodes make up a permissionless blockchain. A consensus method organises transactions, then verifies and constructs the blocks (Singh et al., 2021) – Bitcoin and Ethereum.

Permissioned (Private) blockchain: It is distinct in that nodes are known, recognised and cryptographically validated, and that the number of nodes is allotted to reduce processing time during the consensus process. It also provides authentication to authorised users to read and write using built-in access control and append a block, perform transactions and maintain membership in the blockchain network (Tripathi et al., 2020). As an illustration, consider Corda, HLF and Ethereum.

A consortium or federated blockchain is a collection of public and private blockchains with a decentralised structure. However, consortium blockchain allows multiple organisations to participate rather than a single company managing the entire network. The blockchain allows for dividing organisations based on their use or function. Each organisation is in charge of managing its own identity and controlling who can access it. The ability to choose the nodes

in advance and make the transaction either secret or public is also available (Tripathi et al., 2020).

3.1.3 Hyperledger fabric (HLF)

It is an open-source tool designed to build the decentralised application for blockchain platform, which was initially started by IBM and presently endorses Linux Foundation with licenses including better features and adaptable architecture. To create smart contracts (Guggenberger et al., 2022), it offers generic programming languages (like Node.js, Go and Java) and pluggable consensus protocols (RAFT consensus algorithms are supported by Fabric.). (HF referred to it as chain code) but does not use any cryptocurrency reward structure. The fabric is one of the platforms that perform better in transaction processing and transaction confirmation latency while ensuring that transactions are private and secret (Walia et al., 2022).

Let the blockchain network be represented by a set $N = \{n_1, n_2, ..., n_n\}$ of nodes, where each node ni represents a participant in the network.

$$N = \{n_1, n_2, ..., n_n\} \ (set\ of\ nodes) \tag{1}$$

Let the ledger be represented by a set $L = \{l1, l2, ..., lm\}$ of blocks, where each block lj represents a collection of validated transactions.

$$L = \{l_1, l_2, ..., l_m\} \ (set\ of\ blocks) \tag{2}$$

Let the consensus protocol be represented by a function C (N, L) that governs how nodes in the network agree on the ledger's state. This function inputs the set of nodes N and the set of blocks L and outputs a new block lj+1 that is added to the ledger L.

$$C(N, L) \ (consensus\ protocol) \tag{3}$$

Let the smart contracts be represented by a set $S = \{s_1, s_2, ..., s_k\}$ of programs stored on the blockchain and executed automatically when certain conditions are met. Each smart contract si takes as input a set of parameters P and outputs a set of actions A.

$$S = \{s_1, s_2, ..., s_k\} \ (set\ of\ smart\ contracts) \tag{4}$$

Finally, let the security and privacy of the blockchain be represented by a set of cryptographic functions $F = \{f_1, f_2, ..., f_n\}$ that provide encryption, digital signatures and other cryptographic operations to protect the data and transactions stored on the blockchain.

$$F = \{f_1, f_2, ..., f_n\} \ (set\ of\ cryptographic\ functions) \tag{5}$$

CA (Certificate Authority)*:* All network participants receive certificates from a Certification Authority (CA). The CA has digitally signed these certificates. Fabric CA is a built-in CA component offered by HLF. A distinct node that serves as a certification authority is hosted by each organisation. Mathematically CA is represented as follows:

Let CA be a set of functions and parameters used to manage the digital certificates used to authenticate participants in the network. Let the CA be represented by a tuple (I, G, K, C, R, V),

$$CA = (I, G, K, C, R, V) \tag{6}$$

where:

I: The digital certificates issuer (typically a trusted third party).

G: The set of groups or organisations authorised to participate in the network.

K: The set of cryptographic keys used for encryption, decryption and digital signatures.

C: The set of rules and policies for issuing, validating and revoking digital certificates.

R: The functions and parameters for registering new participants and their digital certificates.

V: The set of functions and parameters for verifying the authenticity and validity of digital certificates.

MSP (Membership Service Provider): To define the constituents of a trusted domain, an MSP determines which Root Certificate Authority and Intermediate Certificate Authorities are trusted. Mathematically, MSP is represented as follows:

Let MSP be a set of functions and parameters used to manage the identities and permissions of participants in the network, respectively. Let the MSP be represented by a tuple (I, G, C, S, V),

$$MSP = (I, G, C, S, V) \tag{7}$$

where:

I: The identity of the MSP (Typically a trusted third party).

G: The set of groups or organisations authorised to participate in the network.

C: The set of rules and policies for managing the identities and permissions of participants.

S: The set of functions and parameters for signing and verifying messages and transactions.

V: The set of functions and parameters for validating the authenticity and permissions of participants.

Organisations: A managed collection of blockchain network users is an organisation. Every organisation uses a Membership Service Provider to manage its members.

Peers: These nodes execute and keep track of the transactions in the ledger. After receiving it as a block from the ordering service, they retain the ordered state update in the ledger.

Clients: They are end users. The peers have received the transaction request that they sent. Additionally, they coordinated the committers and orderers during the verification process.

Orderer: It orders the transactions.

Endorser: They execute the smart contract and simulate the transactions.

Channels: Multiple ledgers shared across network participants (organisations) can be used to execute confidential and private transactions. The term 'channels' refers to these ledgers or private subnets of communication.

Chaincode: In an HLF network, chaincode defines a smart contract. It encodes the rules for particular categories of network transactions using self-executing logic. Each peer on a channel that is taking part must install these chaincodes. It can be called by authorised peers using client-side programs.

3.1.4 IPFS

It is a P2P file system. If one of the nodes in a P2P network, like IPFS, goes down, the other nodes can still deliver the required data. On IPFS, files are content addressed rather than location addressed. Instead of looking for files by location, IPFS searches for files based on the Content Identifier (CID) (Benet, 2014). If one node in a P2P network, like IPFS, goes down, the networks other nodes remain capable of delivering the necessary data. Once a file is uploaded to the IPFS system, it is possible for IPFS to not only store files in a variety of formats but also to retrieve the hash value of the currently open file. When re-accessing the same file, you must utilise the hash code as an index. As an off-chain storage option, this facilitates integration with many blockchains.

3.1.5 Proxy Re-encryption

Re-encryption is a cryptographic method that is also known as proxy re-encryption that enables a third party to convert encrypted data from one encryption key to another without first decrypting and re-encrypting the data. It is used when one party wants to share content that has been encrypted using another party's public key without disclosing its private key. For instance, Bob sent Alice an encrypted message with her public key. Now Alice wants to send Charlie the content without revealing her private key, decrypting it, and re-encrypting it with Charlie's public key. In this instance, Alice performs it using proxy re-encryption. Using Charlie public key and her personal key, Alice may now create a proxy re-encryption key; then choose a proxy for encrypted data to re-encrypt. Charlie can get the stuff that has been re-encrypted by the proxy. Charlie can decrypt data that has been proxy-re-encrypted by his private key (Venkatesan et al., 2021; Manzoor et al., 2019). The necessary outcome is obtained without revealing the content to the proxy at Alice's end and decoding.

3.1.6 ABAC (Attribute-based access control)

This is an access control system that takes attributes, objects, permissions and environments into account when determining access. Its formal definition gives Common Policy: <RL, Opn, Obj, App>; in this, RL stands for the role that the access requester owns, Opn for operation, Obj for information resources and App for application. It assesses if the item possesses the required qualities before deciding whether to provide authorisation. ABAC can offer fine-grained access control, which supports many input decision sets, defines a wide range of potential rules and expresses a wide range of strategies with just modest computational requirements and features (Zhang et al., 2022; Wu et al., 2021). The relationship between the subject and the object may be decoupled with such flexibility. Additionally, ABAC can only be accomplished with additional topic knowledge, negating the necessity to change the laws already in place. Blockchain framework could enhance data privacy and security of devices in intelligent systems by using authentication and access control (Mariam et al., 2021).

3.1.7 Concept of bilinear mapping

The pairing idea is applied to the elliptic curve using the bilinear mapping technique. The symbol e denotes this mapping: $G * G' \to GT$, where G is a Gape-Diffie-Hellman (GDH) group, and GT is another multiplicative cycle group of prime order p that meets specific criteria. The following three properties of the map relation are met and satisfied:

Computability: A good algorithm for computing e should always be available and can be computed efficiently.

Bilinearity: It is represented mathematically as follows: For all a, $b \in Z * p$, for all $P \in G1$, $Q \in G2 : e(aP, bQ) = e(P, Q)^{ab}$

Non-degeneracy: If g is a $G1$ generator, then $e(g, g)$ is a $G2$ generator. This can be expressed as $e(g, g) \neq 1$.

3.2 Methodology

3.2.1 System design

The following are the essential components of our proposed methodology: a hospital administration, a doctor, a patient, an insurance agency and a blockchain consortium called HLF (Androulaki et al., 2018; Monrat et al., 2020; Du et al., 2021) and a distributed file system called IPFS. The encrypted healthcare records files are stored in IPFS (Kumar et al., 2021), and the PEHR created by hospital administration or doctor is stored in HLF. For implementing business logic, chaincode is used for implementing business logic, such as the reading and storage of healthcare records and patient access control strategy, as shown in Figure 2. When there is an emergency, when a patient cannot manage to control their PEHR access rights, with the help of a re-encryption key provided by the planned PEHR system, the doctor who created the record can take action on the patient's behalf. The re-encryption key re-encrypts the patient's encryption key-protected (Babulal and Sharma, 2021) PEHR into a format that the doctor's private key can decipher. One of the challenging difficulties of one-way functions is the discrete logarithm problem, which is used in this study. Given a, $b \in Z * q$, it is challenging for any probabilistic polynomial time intruder I to identify a value of $m \in Z * q$ such that $b = am$, known as the discrete logarithm issue. As a result,

using a public key or ciphertext to acquire the private key is impossible and guarantees the data's privacy and security.

Hospital administration: It is responsible for creating, distributing, and managing digital certificates. Only after receiving certificates may doctors and patients access the HLF. They can create patient EHR data, encrypt it and keep it on third-party storage. They can also seek to delete records and update ones that have already been generated. The patient can provide physicians read and write access on a need-to-know basis. Additionally, by default, each hospital's management department has the right to retrieve the patient's electronic medical record because the patient might not permit the doctor to treat them in an emergency.

Patient: As a patient-centric application, the patient will own all personal and PEHR data. These are the key data owners. After getting the consent of their patients, doctors can read and upload some EMRs. In the same set of ledgers, the EMR is kept for each hospital in the HLF. The patient should be central to the plan for controlling access to the medical record.

Doctor: There are numerous departments in every hospital, and there are numerous doctors in each department. Doctors can access PEHR and upload it after getting patient consent.

Insurance agency: A user or organisation, an individual or entity that accesses data for claims, typically insurance firms.

The hospital is responsible for preparing the PEHR, encrypting it with the patient's public key, and storing it on the cloud or IPFS for smooth operation each time a patient comes in for a consultation. At the same time, a transaction that will be verified and committed into the block must be used to create and post the appropriate patient record's meta-data onto the blockchain.

The following steps describe how our PEHR system process flow, which is built on HLF and the IPFS:

Step 1: Using the ID of department doctors and patients, the hospital administration develops digital certificates for them. Based on the ID, the department generates primary initial data for the patient or doctor in HLF, such as name, age, sex, etc.

Step 2: For an access request, the doctor can make the patient EHR, and the request will either be accepted or rejected by the patient. The healthcare professional may enter the patient's diagnosis information, including any associated images, videos and other materials, into a Patient's Electronic Health Records (PEHR) following diagnosis and treatment.

Step 3: To provide a reliable experience, the PEHR is encrypted by the doctor and uploaded to IPFS while utilising the patient's public key.

Step 4: A distinct hash value based on the file's contents is created by IPFS and sent back to the doctor. The file is subsequently distributed throughout the entire network for storage.

Step 5: Using the client application we used to interact with the Hyperledger Fabric (HLF), the relevant patient record's metadata (multi hash and other pertinent information) must be created and added to the HLF. To add or retrieve the information, execute the chaincode about the files through a transaction, which will then be validated and added to the block.

Step 6: The doctor who wants to access the PEHR sends the client's request for access, and after verification by ABAC on the chaincode, the patient grants or denies the permission.

Step 7: The patient's private key and the doctor's or insurance provider's public key must both be fed into the RENK_GNR function for the patient's record to be shared with the provider of care or insurance to generate the re-encryption key (RENK).

Step 8: To re-encrypt the encrypted record, the patient will give the RENK to a proxy or patient. The allowed encrypted record may be re-encrypted and sent to the appropriate doctor or insurance provider using the appropriate re-encryption key by the patient or proxy.

Step 9: By accessing the HLF's meta-data, a doctor or insurance provider can decrypt the records and confirm their integrity. The healthcare file's hash value is retrieved by the doctor from its metadata and sent to IPFS.

Step 10: Based on the received file hash value index, IPFS collects file blocks from across the whole network and provides them to the client after assembly.

4 PEHR-sharing method for the proposed system

The suggested PEHR-sharing procedure is divided into three phases: user registration, PEHR upload and PEHR sharing. Notations used in our paper for PHER are shown in Table 2.

Table 2 Notations used for PHER in the proposed system

user ID	$UIDN_{pt}$, $UIDN_{dr}$, $UIDN_{in}$
$RENK_{pt}$	Re-encryption Key
ACCPL	Access Policy
$ENPEHR_j$	Encrypted PEHR
$H(PEHR_j)$	Hash value of PEHR
PRTK	Private key
PUBK	Public Key
C_{PEHR}	Cypher Text of PEHR
C_{REPEHR}	Cypher Text of Re-encrypted PEHR

4.1 User's registration phase

This phase focuses on connecting to the HLF blockchain network so that users, including doctors and patients, can manage and share healthcare data. It comprises two phases: identity registration, where the user's identity is registered and authentication, where security parameters are obtained, and an encryption key is generated. The steps are following:

Step 1: The user sends a message to a Certificate Authority (CA) in the Hyperledger fabric network that includes the user's attributes (Name, gender, government ID, etc.) through a Membership Service Provider (MSP). The attributes of the user show whether they are patients or doctors.

Step 2: The CA uses these attributes to determine the user's ID. Once identification is complete, the CA identifies the kind of user based on these attributes to determine whether the legitimate user is available. After that, a random user ID (combination of number and alphabet) is generated based on whether a user is a doctor or a patient and sent to the user along with the certificate. If the user is legitimate, registration is allowed.

Step 3: After successfully registering an account, a user sends the CA a message with the certificate to request security parameters for generating the key of encryption mandatory for PEHR sharing. The CA sends back the security parameters to the user If the certificate is a legitimate user.

Step 4: Users that join the network of Hyperledger create a combination of a private key and a public key utilising the security parameters they receive from the CA. First, the user chooses a random decimal number corresponding to the expression x Є Z*q. The user's secret key is the selected x, which is never revealed. Users then produce a public key for usage in the network using their private key and the key generator. When a PEHR is generated, users who can directly produce the encryption key they use to protect PEHRs can secure it with a unique key every single time.

4.2 Uploading of PEHR

The steps are following:

Algorithm for PEHR (Patient Electronic Health Record) Upload

1. ENCK_GEN($PRTK_{pt}$, PEHR) /* Encryption key generation function*/
   {
   If (PEHR==New PEHR||PEHR==Updated PEHR)
   then Select a random value *R* and generate PEHR encryption key
   $Z^R = e(g, g^R)$
   }
   return Z^R
2. RENK_GNR($PUBK_{dr}$, $PRTK_{pt}$) /* Re-encryption key generation function*/
   {
   If (PEHR==New PEHR||PEHR==Updated PEHR)
   then Re-encryption key ($RENK_{pt\rightarrow dr}$) is generated
   $RENK_{pt\rightarrow dr}$= RENK_GNR($PUBK_{dr}$, $PRTK_{pt}$) = $(g^d)^{1\backslash p} = g^{d/p}$
   return $RENK_{pt\rightarrow dr}$
   else
   return No PEHR is generated
   }
3. ACCPL_GEN(UIDN, PERMISSION, DOCUMENT, PURPOSE) /*Access policy generation function*/
   {
   user_attributes:= {
   "Doctor": {"UIDN": $UIDN_{dr}$, "PERMISSION": read&&write, "DOCUMENT": PEHR, "PURPOSE": Treatment},
   "Patient": {"UIDN": $UIDN_{pt}$, "PERMISSION": Grant&&Deny, "DOCUMENT": PEHR, "PURPOSE": Treatment}
   "Insurer": {"UIDN": $UIDN_{in}$, "PERMISSION": read, "DOCUMENT": PEHR, "PURPOSE": Claim}
   }
   user:= user_attributes[input.user]
   user: UIDN == "Doctor" or "Patient" or "Insurer"
   If (UIDN = =$UIDN_{pt}$) **then**
   If (permission=="GRANT" && UIDN== "$UIDN_{dr}$") **then**
   Write or read the health data from or to the specified $UIDN_{pt}$ PEHR
   Create patient centric view of PEHR in IPFS
   Elseif
   If (permission== "GRANT" && UIDN== "$UIDN_{in}$") **then**
   Read PEHR for claiming
   Else
   Permission=Deny
   }
   return ACCPL
4. The patient sends the doctor who treated him or her encryption keys (Z^R), $RENK_{pt\rightarrow dr}$, ACCPL and $UIDN_{pt}$.
5. ENC(Z^R, PEHR)
   {
   If(PEHR==New PEHR||PEHR==Updated PEHR)
   $C_{PEHR} = (Z^R * PEHR, g^{RPRTK_{pt}})$
   return C_{PEHR}
   else
   return No PEHR generated
6. **If** (PEHR== C_{PEHR})
   $H(C_{PEHR})$ ← Upload C_{PEHR} to IPFS
7. Metadata ← {$UIDN_{dr}$, $H(C_{PEHR})$, Timestamp}
   TRANSACTION ← {$UIDN_{pt}$, $UIDN_{dr}$, DS_{dr}, Metdata, ACCPL, Timestamp}
   HyperledgerNetwork ← Upload TRANSACTION

4.3 Algorithm for PEHR sharing

1) The doctor searches those transactions by executing a chaincode where the information of desired patient electronic health records is stored.
2) The chaincode searches for a transaction that contains the required information and returns it using the user ID.
3) By using the specified hash value in the transaction from the InterPlanetary File System, the encrypted patient electronic medical record CPEHR downloaded by the doctor who gets the information of the transaction.
4) The patient, whose encrypted patient electronic medical record is this, is asked for the re-encryption key $RENK_{pt\rightarrow dr}$ by the doctor, who then requests its re-encryption by executing the re-encryption key request chaincode.

5) The chaincode that sends the request for the re-encryption key verifies the requested user's authenticity according to the patient's access policies.

6) The re-encryption request is sent to the patient through the chaincode if the user's authenticity is satisfied according to the access policy. Otherwise, the request may be denied.

7) In response to the message asking for the re-encryption key, the patient generates $RENK_{pt \rightarrow dr}$ using the doctor's public key and sends it to the doctor.

8) If a patient is not responding in the case of an unresponsive state, the emergency event chaincode executes. A message asking for the re-encryption key is sent to the authorised hospital via the chaincode. In response to the message asking for the re-encryption key, the authorised hospital generates $RENK_{hosp \rightarrow dr}$ using the doctor's public key and sends it to the doctor.

9) The doctor who receives the re-encryption key $RENK_{pt \rightarrow dr}$ utilises it to convert CPEHR into CREPEHR by re-encrypting it.

10) The doctor decrypts C_{REPEHR} using their private key to get the original PEHR

$DEC(C_{REPEHR}, PRTK_{Pr})$ /* Decryption function

{

$PEHR \leftarrow DEC(C_{REPEHR}, PRTK_{Pr}) = Z^R * C_{PEHR}/e(g, g^{rd})^{1/d} = Z^R * C_{PEHR} / Z^R$

}

return PEHR

---

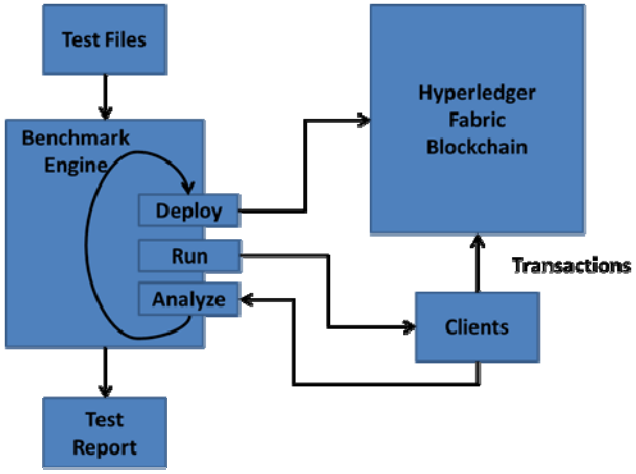
5 Result and discussion

HLF was independently deployed, and necessary components were set up, including four CouchDB instances and two organisations (Org1 and Org2), each with two peers (one committing peer and one endorsing peer). The ordering service used the RAFT consensus process. The HLF LTS version (Fabric 2023a, 2023b) performance was the main focus of our study. The tests were conducted using an Ubuntu 18.04 LTS operating system, a 3.40 GHz Intel Core i5-3570 processor, 16 GB of RAM and 500 GB of disc space. We have used GO language for implementing the chaincode.

The performance of our proposed network is evaluated using Hyperledger Caliper (Caliper, 2023a, 2023b). It enables users to test various blockchain systems with specified use cases and parameters. The Hyperledger community has made it available for evaluating the efficiency of blockchain systems and generating reports with good metrics like throughput and latency. It can create a workload for a System Under Test (SUT) and continually track the results from this SUT.

As shown in Figure 3, the configuration details include the benchmark that needs to be run, the blockchain framework that needs to be tested and the smart contract code. Two different types of clients, Master and Local clients are produced for feeding these into the interface. The System Under Test (SUT) is configured using the Master client. Channels are formed, peers can join channels and chaincode is deployed using the master client. A loop is started by the master client to run tests in accordance with the benchmark configuration file. According to the predetermined workload, tasks will be generated and allocated to local clients. The local clients' returned performance statistics will be saved for later analysis. A report is generated automatically when statistics from all clients of each test round have been analysed.

Figure 3 Testing methodology for Hyperledger Caliper



We crafted two experiments to evaluate the HLF's performance and IPFS integrated health record-sharing system. It is possible to assess performance and scalability using the workload and number of nodes as variables. The number of transactions and concurrent requests from the various numbers of nodes (2, 20 and 40) are included. Within our proposed system, two functions, 'adding PEHR' and 'query PEHR', were implemented to upload the PEHR and create a ledger query.

5.1 Evaluation metrics

We have considered the below matrices to evaluate the performance of our proposed method approach.

5.1.1 Transaction throughput

It is the number of transactions completed per second. Let T be the total number of transactions processed by the system in a given time frame, and let Δt be the duration of the time frame. Then, the Transaction Throughput (TP) can be calculated as follows:

$$TP = T / \Delta t \tag{8}$$

The transaction throughput can be measured in Transactions Per Second (TPS) or Per Minute (TPM).

5.1.2 Average latency

This is the average interval of time between the transaction's initialisation and the transaction's actual execution.

Let L_i be the latency of the i-th transaction, and let n be the total number of transactions processed by the system. Then, the Average Latency (AL) can be calculated as follows:

$$AL = (1/n) * \Sigma(L_i) \quad (9)$$

The average latency can be measured in seconds or milliseconds.

Figures 4 and 5 depict the transaction throughput after adding and querying PEHR functions, which are executed on 50 to 1000 transactions each. The three lines depict the creation and querying operations for three peer nodes (i.e., 2, 20 and 40 nodes). It has been noted from the figure that the query function's transaction throughput is higher than the adding PEHR functions. As the transaction generates up to 500, nominal growth is detected in the adding and the query function for 2 and 20 peer nodes in both functions and continuous degradation in 40 peer nodes in the creating function. However, it is steady in the case of the query function.

Figure 4 Transaction throughputs on adding PEHR function (see online version for colours)

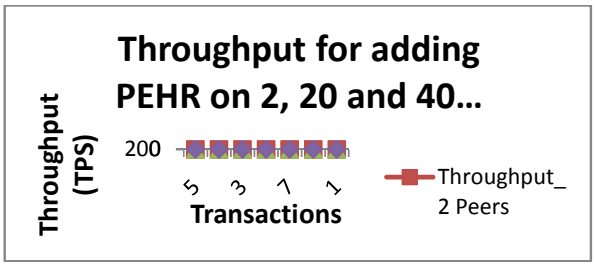
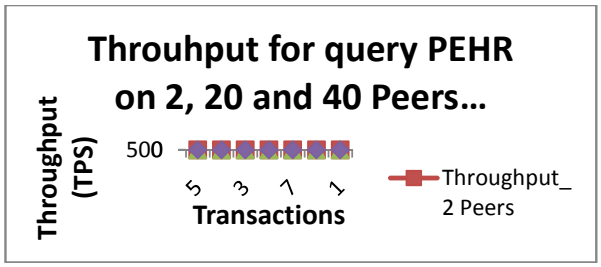


Figure 5 Transaction throughputs on query PEHR function (see online version for colours)



Figures 6 and 7 shows the average transaction latency after adding and querying PEHR functions and executed on 50 to 1000 transactions each. It is noticed from the three lines in the graph that as both functions are processing more transactions, the average transaction latency is continuously growing. The latency of adding PEHR is more as compared to the query function. From Figures 5 and 6, it is concluded that with fewer nodes, the throughput is more and latency is low. The throughput of both functions diminishes and the delay raises as the number of nodes rises.

Figure 6 Transaction latency on adding PEHR function (see online version for colours)

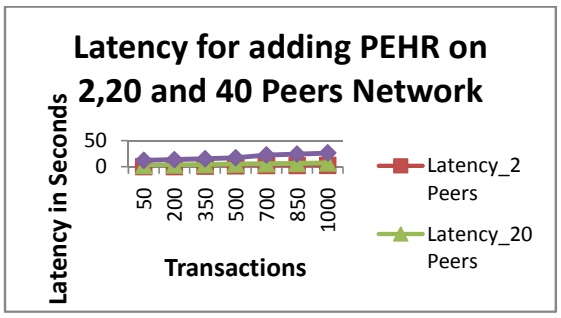


Figure 7 Transaction latency on query PEHR function (see online version for colours)

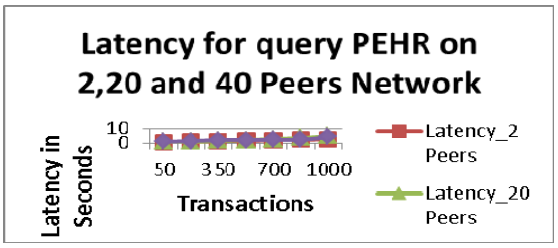
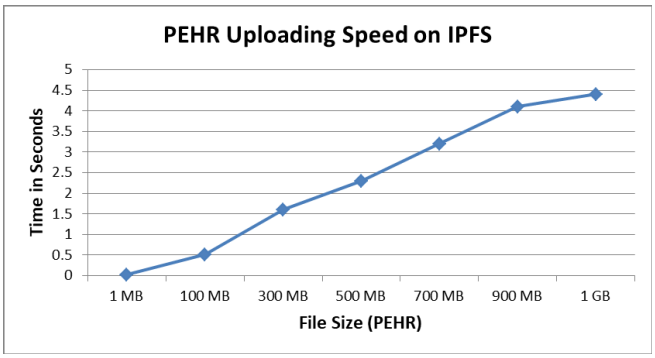
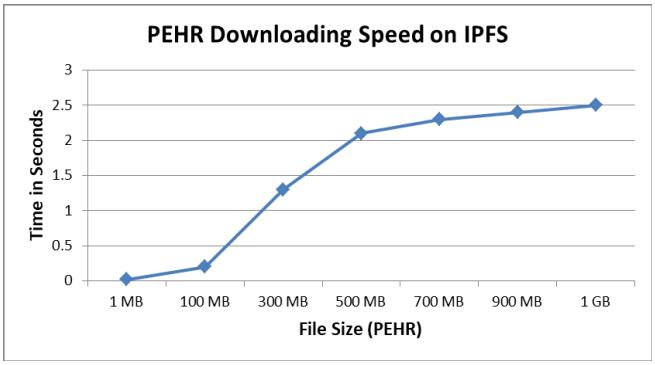


Figure 8 PEHR uploading speed on IPFS



To test IPFS uploading and downloading, we have considered varying file sizes from 1 MB to 1 GB. It is observed from Figures 8 and 9 that as the size of the healthcare data file increases; the overall time slightly increases in uploading and downloading the file. Although IPFS may not be the optimal database for applications, it does serve the function of preserving files through hashing and providing access control via HLF.

Figure 9 PEHR downloading speed from IPFS



By analysing the system's performance using these metrics, we may determine the system's advantages and disadvantages and then adjust its design to perform better in actual situations. This evaluation can be conducted under varying conditions, such as different network loads, numbers of nodes and security and privacy requirements, to determine the system's scalability and robustness. The transaction throughput metric indicated that the system could process many patient EHR transactions within a short period, demonstrating its efficiency. Meanwhile, the average latency metric showed that the system could quickly store and retrieve patient EHR data, reducing patient and healthcare providers' waiting time.

6 Conclusions

This study introduced the PEHR system, which enables patients to manage their medical data and is built on a Consortium Blockchain (CB). By addressing the issues with the current blockchain-based medical system, the PEHR system can maintain and share PEHRs securely. Scalability and privacy problems plaguing blockchain-based medical systems are resolved by the PEHR system using a distributing technique of data sharing and a structure of the lightweight transaction. Due to the lightweight transaction structure, much data can be stored in the block as these blocks hold only the tiniest information, such as the PEHR summary data and encrypted PEHR metadata. The re-encryption-based data encryption approach solves the issue of data leaking and the theft of private information when exchanging PEHRs. Smart contracts were employed in the PEHR-sharing procedure. Smart contracts implemented security level-based access control and stopped unauthorised users from accessing healthcare data.

Changes in transaction and node count in the blockchain network were the main subjects of the analysis. A study was done on the effects of scaling up to 40 nodes and altering the workload up to 1000 transactions. Throughput and average latency are among the parameters used for evaluation. With the workload increasing and the number of nodes increasing to 20, the transaction throughput is consistent. Once the workload is increased to 1000 transactions across 40 nodes, the throughput drops and is still erratic. The workload and the number of nodes affect latency, which rises as a result. For 20 nodes with 1000 transactions, it does, however, decrease. As a result, we conclude that the PEHR blockchain network is best suited for a small consortium network and needs to grow to support many nodes when the volume of transactions is high. Moreover, IPFS may not be the optimal database for applications; it does serve the function of preserving files through hashing and providing access control via HLF. In the future, we shall improve our work for large-scale nodes.

References

Abunadi, I. et al. (2019) 'BSF-EHR: blockchain security framework for electronic health records of patients', *Sensors*, No. 8. Doi: 10.3390/s21082865.

Alder, S. (2023) *Study on the Cost of Data Breaches Reveals Rising Costs in US healthcare: The HIPAA Journal*. January 2023 Healthcare Data Breach Report. https://www.hipaajournal.com/january-2023-healthcare-data-breach-report/ (accessed on 31 March 2023).

Alkouz, A. et al. (2021) 'EPPR: blockchain for educational record sharing and recommendation using the Ethereum platform', *International Journal of Grid and Utility Computing*, Vol. 12, No. 3, pp.347–356.

Andriamanalimanana, B. et al. (2020) 'Efficient variant transaction injection protocols and adaptive policy optimisation for decentralised ledger systems', *International Journal of Grid and Utility Computing*, Vol. 11, No. 6, pp.847–856.

Androulaki, E. et al. (2018) 'Hyperledger fabric: a distributed operating system for permissioned blockchains', *Proceedings of EuroSys Conference*, ASM, New York, USA.

Babulal, C. and Sharma, A. (2021) 'Modified DES cryptosystem with steganography for healthcare systems in IoT', *Design Engineering*, pp.5530–5538.

Benet, J. (2014) *Ipfs – content addressed, versioned, p2p file system*. Available online at: https://arxiv.org/abs/1407.3561v1

Caliper, H. (2023a) *Hyperledger Caliper Architecture*, Electronic Article. Available online at: https://hyperledger.github.io/caliper/ (accessed on 9 February 2023).

Caliper, H. (2023b) *Hyperledger caliper architecture.* Available online at: https://www.hyperledger.org/projects/caliper (accessed on 9 February 2023).

Chen, Y. et al. (2021) 'A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection', *Wireless Comm. and Mobile Comp P*, No. 1/12. Doi: 10.1155/2021/6685762.

Da Silva Costa T.B. et al. (2022) 'Blockchain-based architecture design for personal health record: development and usability study', *The Journal of Medical Internet Research*, Vol. 24, No. 4. Doi: 10.2196/35013.

Du, M., Chen, Q., Chen, J. and Ma, X. (2021) 'An optimized consortium blockchain for medical information sharing', *IEEE Transactions on Engineering Management*, Vol. 68, No. 6, pp.1677–1689.

Fabric, H. (2023a) *Hyperledger Fabric*, Electronic Article. Available online at: https://hyperledger-fabric.readthedocs.io/en/release-2.5/ (accessed on 9 February 2023).

Fabric, H. (2023b) *Hyperledger Fabric*. Available online at: https://www.hyperledger.org/use/fabric (accessed on 9 February 2023).

Gorenflo, C. et al. (2020) 'FastFabric: scaling hyperledger fabric to 20,000 transactions per second', *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Vol. 30. Doi: 10.1109/BLOC.2019.8751452.

Guggenberger, T. et al. (2022) 'An in-depth investigation of the performance characteristics of hyperledger fabric', *Computers and Industrial Engineering*, Vol. 173. Doi: 10.1016/j.cie.2022.108716.

Herwanto, R. et al. (2021) 'Measuring throughput and latency distributed ledger technology: Hyperledger', *Journal of Information Technology Ampera*, Vol. 2, No. 1, pp.17–31.

Huang, J. et al. (2022) 'Sharing medical data using a blockchain-based secure EHR system for New Zealand', *IET Blockchain*, Vol. 2, pp.13–28.

Ismail, L. et al. (2020) 'Performance evaluation of a patient-centric blockchain-based healthcare records management framework', *Proceedings of the 2nd International Electronics Communication Conference*, ACM, New York, USA, pp.39–50.

Jain, A. and Jat, D.S. (2021) 'Implementation of blockchain enabled healthcare system using Hyperledger fabric', *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence*, ACM, New York, USA, pp.37–47.

Kumar, R., Tripathi, R., Marchang, N., Srivastava, G., Gadekallu, T.R. and Xiong, N.N. (2021) 'A secured distributed detection system based on IPFS and blockchain for industrial image and video data security', *Journal of Parallel and Distributed Computing*, Vol. 152, pp.128–143.

Li, L. et al. (2021) 'Electronic medical record sharing system based on hyperledger fabric and inter planetary file system', *Proceedings of the 5th International Conference on Computing and Data Analytics (ICCDA'21)*, ACM, New York, USA, pp.149–154.

Liu, J. et al. (2020) 'A privacy-preserving medical data sharing scheme based on consortium blockchain', *IEEE Global Communications Conference*, Taiwan, pp.1–6.

Liu, T. et al. (2020) 'A privacy-preserving medical data sharing scheme based on consortium blockchain', *IEEE Global Communications Conference*, Taipei, Taiwan, pp.1–6.

Mani, V. et al. (2021) 'Hyperledger healthchain: patient-centric IPFS-based storage of health records', *Electronics*, Vol. 10, No. 23. Doi: 10.3390/electronics10233003.

Manzoor, A. et al. (2019) 'Blockchain based proxy re-encryption scheme for secure IoT data sharing', *IEEE Proceedings of the International Conference on Blockchain and Cryptocurrency*, Seoul, South Korea, pp.99–103.

Mariam, S. et al. (2021) 'Enhanced authentication and access control in internet of things: a potential blockchain-based method', *International Journal of Grid and Utility Computing*, Vol. 12, Nos. 5/6, pp.469-485.

Monrat, A.A. et al. (2020) 'Performance evaluation of permissioned blockchain platforms', *IEEE Asia-Pacific CCSDE*, Gold Coast, Australia, pp.1–8.

Nakamoto, S. (2008) *Bitcoin: A Peer-to-Peer Electronic Cash System*. Available at http://dx.doi.org/10.2139/ssrn.3440802

Nasir, Q. et al. (2018) 'Performance analysis of Hyperledger fabric platforms', *Security and Communication Networks*. Doi: 10.1155/2018/3976093.

National Institutes of Health (2023) *National Institutes of Health*. Available online at: https://www.nih.gov/ (accessed on 31 March 2023).

Nguyen, D.C. et al. (2019) 'Blockchain for secure EHRs sharing of mobile cloud based e-health systems', *IEEE Access*, Vol. 7, pp.66792–66806.

Nyaletey, E. et al. (2019) 'BlockIPFS – blockchain-enabled interplanetary file system for forensic and trusted data traceability', *IEEE International Conference on Blockchain (Blockchain)*, IEEE, USA, pp.18–25.

Pradhan, N.R. et al. (2022) 'A novel blockchain-based healthcare system design and performance benchmarking on a multi-hosted testbed', *Sensors*, Vol. 22, No. 9. Doi: 10.3390/s22093449.

Roehrs, A. et al. (2019) 'Analyzing the performance of a blockchain-based personal health record implementation', *Journal of Biomedical Informatics*, Vol. 92. Doi: 10.1016/j.jbi.2019.103140.

Sharma, A. et al. (2022) 'Blockchain-based internet of things (IoT) for healthcare systems: COVID-19 perspective', *Healthcare Monitoring and Data Analysis using IoT: Technologies and Applications*, Vol. 38, p.355.

Sharma, V. and Lal, N. (2020) 'A detail dominant approach for IoT and blockchain with their research challenges', *Proceedings of the International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, Lakshmangarh, Sikar, India, pp.1–6. Doi: 10.1109/ICONC345789.2020.9117533.

Sharma, V. and Lal, N. (2022) 'Role of blockchain in IoT enabled power and energy related healthcare system-platform for the development of IoT security', in Malik, H., Ahmad, M.W. and Kothari, D. (Eds): *Intelligent Data Analytics for Power and Energy Systems*, Vol 802, Springer, Singapore. Doi: 10.1007/978-981-16-6081-8_27.

Shuaib, K. et al. (2022) 'Secure decentralized electronic health records sharing system based on blockchains', *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, No. 8, pp.5045–5058.

Singh, C. et al. (2021) 'Medi-block record: secure data sharing using block chain technology', *Informatics in Medicine Unlocked*, Vol. 24. Doi: 10.1016/j.imu.2021.100624.

Sun, J. et al. (2020) 'A blockchain-based framework for electronic medical records sharing with fine-grained access control', *PLoS One*, Vol. 15. Doi: 10.1371/jrnl.pone.0239946.

Tripathi, G. et al. (2020) 'S2HS – a blockchain based approach for smart healthcare system', *Healthcare*, Vol. 8, No. 1. Doi: 10.1016/j.hjdsi.2019.100391.

Venkatesan, S. et al. (2021) 'Secure and decentralized management of health records', in Namasudra, S. and Deka, G.C. (Eds): *Applications of Blockchain in Healthcare Studies in Big Data*, Singapore, Springer, Vol. 83, pp.115–139.

Verma, V. and Sharma, A. (2022) 'Analysis and classification of security mechanisms on the cloud for digital healthcare', *Proceedings of the 11th International Conference on System Modeling and Advancement in Research Trends*, Moradabad, India, pp.1596–1601.

Walia, V. et al. (2022) 'Blockchain in IoT and limitations', *Trust-Based Communication Systems for Internet of Things Applications*, pp.17–27.

Wen, X. (2023) 'Application of blockchain technology in copyright protection of digital music information', *International Journal of Grid and Utility Computing*, Vol. 14, Nos. 2/3, pp.136–145.

Wu, H. et al. (2021) 'Security and privacy of patient information in medical systems based on blockchain technology', *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 17, pp.1–17.

Xi, P. et al. (2022) 'A review of blockchain-based secure sharing of healthcare data', *Applied Sciences*, Vol. 12, No. 15. Doi: 10.3390/app12157912.

Yazdinejad, A. et al. (2020) 'Decentralized authentication of distributed patients in hospital networks using blockchain', *IEEE Journal of Biomedical and Health Informatics*, Vol. 24, No. 8, pp.2146–2156.

Zhang, D. et al. (2022) 'A secure and privacy-preserving medical data sharing via consortium blockchain', *Security and Communication Networks*. Doi: 10.1155/2022/2759787.

Zhang, Y. et al. (2022) 'Blockchain-enabled decentralized attribute-based access control with policy hiding for smart healthcare', *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, No. 10, pp.1319–1578.