Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Secure decentralized electronic health records sharing system based on blockchains

Khaled Shuaib [a,*], Juhar Abdella [b], Farag Sallabi [a], Mohamed Adel Serhani [a]

[a] College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates
[b] Royal Melbourne Institute of Technology, Melbourne, Australia

ARTICLE INFO

Article history:
Received 16 January 2021
Revised 21 April 2021
Accepted 8 May 2021
Available online xxxx

Keywords:
Electronic health records
Blockchain
Smart contracts
Privacy
Decentralized file system
Covid-19

ABSTRACT

Blockchain technology has a great potential for improving efficiency, security and privacy of Electronic Health Records (EHR) sharing systems. However, existing solutions relying on a centralized database are susceptible to traditional security problems such as Denial of Service (DoS) attacks and a single point of failure similar to traditional database systems. In addition, past solutions exposed users to privacy linking attacks and did not tackle performance and scalability challenges. In this paper, we propose a permissioned Blockchain based healthcare data sharing system that integrates Blockchain technology, decentralized file system and threshold signature to address the aforementioned problems. The proposed system is based on Istanbul Byzantine Fault Tolerant (IBFT) consensus algorithm and Interplanetary File System (IPFS). We implemented the proposed system on an enterprise Ethereum Blockchain known as Hyperledger Besu. We evaluated and compared the performance of the proposed system based on various performance metrics such as transaction latency, throughput and failure rate. Experiments were conducted on a variable network size and number of transactions. The experimental results indicate that the proposed system performs better than existing Blockchain based systems. Moreover, the decentralized file system provides better security than existing traditional centralized database systems while providing the same level of performance.

1. Introduction

Because of life circumstances and the need to get alternative treatments from various care providers, patients often visit different doctors and hospitals throughout their lives. The study by Rouhani et al. (2019) indicated that the average patient in the US sees 18.7 different doctors and has approximately 19 unique medical records during his life time. This figure does not include general healthcare entities such as pharmacies and physiotherapy centers. In addition, patients' data can also be collected from various types of smart devices such as wearable devices and other data entered manually by patients. As a result, in toady's healthcare system, patients' data is scattered in different locations, probably duplicated, mismatched and fragmented across various healthcare providers including public and private hospitals, insurance companies, senior citizens care providers, pharmacies and government regulatory bodies (Tanwar et al., 2020). In order to provide best healthcare services, care providers need to communicate with each other regularly to collect, share and process data that resides in these different locations (Al-Karaki et al., 2019). Moreover, advances in medical research is unimaginable without researchers getting access to the necessary patients' data. However, data sharing between various stake holders is currently very challenging due to the lack of standardized data formats, reliable communication media, interoperability issues and secure platforms to guarantee users' security and privacy.

In a traditional healthcare system, when patients want to share their data with other parties such as hospitals or research institutions, they are required to go through a manual consent process that is very inefficient for care providers to coordinate, specifically in situations where a patient might geographically relocate without knowing in advance where would he be receiving treatment (Dubovitskaya et al., 2017). Moreover, the process of transferring

* Corresponding author.
E-mail addresses: k.shuaib@uaeu.ac.ae (K. Shuaib), S3753266@student.rmit.edu.au (J. Abdella), f.sallabi@uaeu.ac.ae (F. Sallabi), serhanim@uaeu.ac.ae (M.A. Serhani).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.05.002
1319-1578/

Please cite this article as: K. Shuaib, J. Abdella, F. Sallabi et al., Secure decentralized electronic health records sharing system based on blockchains, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2021.05.002

patient's data from one institution to another is time consuming and administratively troublesome (Rahman et al., 2019).

On the other hand, security and privacy breaches associated with healthcare systems and data is also becoming increasingly important to address. The study in Al-Karaki et al. (2019) stated that patients' data is being sold by hackers at a price of up to 20 times higher than that of banking data. Attacks on health databases have resulted in a loss of nearly $30 billion in the past 20 years alone (Al-Karaki et al., 2019). According to the data breach report published by HIPAA Journal (Journal, 2020), more than 268 million healthcare records of US citizens have been compromised between 2009 and 2020 which is equivalent to more than 81% of the population. On the other hand, IBM data breach survey (IBM, 2020) revealed that the average cost of data breaches incurred by healthcare companies in 2020 was 7.13 million which is 10% more than that of 2019.

Because of these concerns, recently Blockchain technology has been proposed by several researchers as a solution to many of the aforementioned problems (Azaria et al., 2016; Daraghmi et al., 2019; Rouhani et al., 2019; Roehrs et al., 2017; Yue et al., 2016; Zheng et al., 2018; Ichikawa et al., 2017; Dubovitskaya et al., 2017; Xu et al., 2019; Rahman et al., 2019; Xia et al., 2017). Blockchain technology can provide several benefits for healthcare systems including a decentralized infrastructure, allowing interoperability, security, authentication, and integrity (McGhin et al., 2019). It relies on a decentralized infrastructure which is more secure, resistant to a single point of failure and has no communication bottlenecks like that of a centralized system. It is based on a hashed linked list data structure that provides tamper proof and immutable data storage ensuring the integrity of health data. Apart from that, it allows for the implementation of a secure access control and permission management scheme for sharing data via secure computer programs known as smart contracts. Moreover, research indicates that a Blockchain based system would also reduce the cost of data maintenance and sharing (Kumar et al., 2020; Zhang et al., 2018). Besides the many benefits Blockchains could provide for healthcare systems, it can also be vulnerable to some security and privacy risks such as the loss of private key, smart contract vulnerabilities and transaction privacy leakages (Li et al., 2020) that might lead to the theft or destruction of patients' data. Therefore, Blockchains should not be considered as a bulletproof platform that can solve all the security problems and necessary design considerations should be taken while applying a Blockchain based solution for a sensitive application like healthcare systems.

The majority of previous studies mainly proposed Blockchains for interoperability, permission management, integrity and privacy. For example, some of the most prominent works in this area (Azaria et al., 2016; Roehrs et al., 2017; Daraghmi et al., 2019; Dubovitskaya et al., 2017) emphasize the aforementioned functions. To that end, almost all existing works utilize Blockchains as a permission management platform leveraging the power of smart contracts (Azaria et al., 2016; Daraghmi et al., 2019; Rouhani et al., 2019; Roehrs et al., 2017; Yue et al., 2016; Zheng et al., 2018; Ichikawa et al., 2017; Dubovitskaya et al., 2017; Xu et al., 2019; Xia et al., 2017). Other studies also attempted to address scalability issues (Roehrs et al., 2017). Except for few studies which proposed their own Blockchains platforms (Roehrs et al., 2017; Yue et al., 2016), existing research mainly relies on Hyperledger Fabric (Androulaki et al., 2018) and Ethereum (Buterin, 2014) as their underlying Blockchains platforms.

However, there are some important limitations in the existing literature. The design of some of the well known previous studies (Azaria et al., 2016; Daraghmi et al., 2019; Dubovitskaya et al., 2017) stores health data in a single location such as the providers' database. This type of approach is not significantly different from a centralized system when it comes to a DoS attack and communication bottleneck. Although a Blockchain based permission management system is decentralized, the data storage and access is still centralized. Thus, it is vulnerable to any of the security threats that exist in a centralized system. The other limitation is that several studies store mapping data that shows the relationship between patients and care providers in smart contracts on the Blockchain. However, this can expose the user to privacy linking attacks as attackers can link user's physical activities with the mapping data on the Blockchain (MedRec, 2020).

Besides this, existing studies have performance and scalability challenges that slow the time of data access (Mazlan et al., 2020; McGhin et al., 2019). Healthcare requires fast access to data, for example in emergency cases, where the patient is not in a state to give permission (e.g. stroke or heart attack) and needs urgent treatment from physicians and surgeons (Kumar et al., 2020). To the best of our knowledge, there is no past studies which consider this issue. As such, many of the Blockchain papers proposed so far would be considered somewhat impractical. Lastly, only few studies conducted a partial implementation and performance evaluation of Blockchain based systems (Daraghmi et al., 2019; Thwin and Vasupongayya, 2020; Roehrs et al., 2019). However, some of the key performance indicators such as end to end latency and throughput as well as scalability can be further studied.

To fill these gaps, we propose a permissioned Blockchain based healthcare data sharing system where a set of pre-determined entities take the responsibility of mining and validating transactions whereas ordinary users need to register to join the network. The proposed system relies on Istanbul Byzantine Fault Tolerant (IBFT) consensus algorithm and an enterprise Ethereum Blockchain platform known as Hyperledger Besu. We believe that this type of Blockchain architecture is more appropriate for enterprise applications like a healthcare system as entities such as hospitals and government regulatory bodies are already taking the responsibility of creating, maintaining and regulating associated data. The proposed system uses Blockchains for permission management while storing the actual data off chain in a secure decentralized file system known as Interplanetary File System (IPFS) (Benet, 2014). This makes the system robust against security attacks such as a DoS attack and integrity attacks as compared to other solutions which maintain the data in a single location. Moreover, it allows data access from multiple locations. To protect the privacy of the user, only a hashed reference index to the off chain data is stored on the Blockchain. The proposed system also employs a threshold signature scheme for recording data on the Blockchain in order to protect the privacy of users from linking attacks that arise from associating patients with each provider they have visited. In addition to that, we perform a detailed performance evaluation based on several performance metrics such as latency, throughput and failure rate. The contributions of this paper include:

- A Proposed permissioned Blockchain based decentralized electronic health record sharing architecture and smart contract design that provides better performance in terms of latency, throughput and success rate as compared to existing solutions. The proposed system relies on the IBFT algorithm and an enterprise Ethereum Blockchain platform known as Hyperledger Besu.
- Integration of a decentralized file system for off-chain data storage that provides comparable performance to existing centralized database systems while providing better security which protects the system from DoS attacks, single point of failure and improves data integrity.
- Provides a detailed implementation and performance evaluation of Blockchain based healthcare system.

This paper is organized as follows. We review related work in Section 2. The proposed Blockchain based health data sharing system is presented in Section 3. In Section 4 the proposed system operational details are presented. We discuss implementation and performance evaluation metrics in Sections 5 and 6. Experimental results and related discussions are presented in Section 7. Section 8 summarizes and concludes the paper.

2. Related work

One of the early work to introduce Blockchain-based medical health record sharing is Medrec (Azaria et al., 2016). Medrec uses Blockchain as a permission management platform to enable data sharing among different entities such as hospitals, insurance companies, pharmacies and patients. The data is neither transferred over the Blockchain nor is stored somewhere in a public location. Instead, it resides at the source location where it is created. Data requesters contact the data holder offline after being granted the necessary permission on the Blockchain network. Medrec was proposed to address four main challenges of traditional healthcare systems: fragmented data access, interoperability; improved data sharing and patient participation problems. It utilizes three different types of smart contracts to achieve its goals: A registry contract that serves as permission database, a summary contract that stores the summary of users data across various care provides and a patient's provider contract that keeps the mapping between each patient and care providers.

However, Medrec has some important limitations. First, the data is still maintained by individual hospitals and pharmacies which makes it susceptible to DoS attacks. The security of data is guaranteed by a single data holder similar to traditional centralized systems with no redundancy. If that single data holder is compromised, then the data could also be compromised. Moreover, the single data holder could become a bottleneck as several data requesters might contact that single data holder simultaneously. In addition to that, even though all metadata belonging to a specific user exists on the Blockchain, data requests still have to fetch the actual data offline from multiple locations which makes it inefficient. Second, Medrec relies on a Blockchain consensus protocol known as Proof of Work (PoW) which does not only consume huge amount of energy but also has high latency and low throughput. Third, the design of Medrec creates additional privacy risks as the care provider submits transactions to the Blockchain using the same Blockchain address. Lastly, Medrec does not allow fine grained permissions. An entity having a permission to access a given data can access the data indefinitely without limits. Moreover, there is no different levels of permission access such as view only, perform computation or download. For example, unlike doctors, researchers might only need to perform statistical calculation on data and not be allowed to view raw data.

Daraghmi et al. (2019) proposed a new architecture for Blockchain based electronic medical system which aims at providing interoperablity, security, and fast access to data while preserving the privacy of patients. This work also suggests a new incentive mechanism that is based on providers' contribution of valid medical records to motivate providers to participate in the mining process. This work applies permissions revoking based on smart contracts. It also employs a proxy re-encryption service to transfer encrypted health records among provider nodes with no need to share the symmetric key used to encrypt/decrypt the message. The performance of the system was evaluated by studying the effect of the number of submitted queries and the volume of medical records in providers databases on the average response time, throughput and communication overhead. However, the conducted experiment focuses only on query transactions and there is no further provided details about the size of the network considered for the experiment. One of the limitation of the proposed approach is that it provides the stewardship of patients and their data to a provider node. For example, a patient is statically bound to a specific provider node and the patient's data is also owned by the provider. The patient has to be registered on the Blockchain via the provider node. In reality, patients do not belong to a specific provider. Rather, they can have relationships with multiple provides at any time and also should be given the full ownership of their data. This type of design also creates extra steps and storage in record adding, retrieving and access control checks. Moreover, maintaining provider-patient mappings on a public smart contract like that of Azaria et al. (2016) could expose the patient's privacy to linking attacks. In addition, as is the case for work done by others, the data is still located in a single location which makes it prone to DoS attacks. Lastly, the study assumes that patients also run Blockchain nodes similar to providers which seems infeasible considering the limited resource and technical knowledge ordinary users have.

MediChain™(Rouhani et al., 2019) provided an implementation of a Hyperledger based Blockchain system for the purpose of managing medical data assets. The system consists of three components: an access control module, an off-chain data storage and a user interface. Medical data is stored off-chain in encrypted format on the cloud while the hash of the data is placed on the Blockchain. However, the system does not consider interoperability.

The authors of Rahman et al. (2019) proposed a Blockchain handshaking mechanism to integrate existing cloud based electronic health record management system with a public Blockchain with the aim of preserving the integrity of healthcare data. The authors have provided a prototype implementation to as a proposed concept for the proposed approach.

The work by Tanwar et al. (2020) proposed a permissioned Blockchain-based electronic health record system that enables data sharing between patients and care providers such as clinicians and laboratories. The proposed system assumes a fully trusted network where all participants are known to each other. Entities participating in the network are registered by the system based on a membership service certificate authority and managed by a network administrator. This paper is one of the first to consider performance evaluation of a Blockchain-based healthcare system. Several performance metrics such as latency and throughput, memory usage, CPU and network consumption have been considered. The experiment was done on a network that consists of two organizations each having two peers. However, there are some key limitations to this work. The proposed system is indistinguishable from a centralized system as the administrator has control over everything including generating private keys for users. Moreover, the system relies on a Hyperledger fabric which uses the RAFT consensus protocol and does not tolerate node security failures. In addition to that, the experiment was done on a number of peers and the effect of varying the number of peers on latency and throughput was not considered.

OmniPHR (Roehrs et al., 2017; Roehrs et al., 2019) proposed a distributed computing platform that provides interoperability for personal health records in different organizations based on OpenEHR standard (openEHR, 2020). The authors suggested a new type of system that consists of a network of servers performing a similar function as permissioned Blockchain nodes. The servers are responsible for validating, distributing and storing data blocks submitted by users and care providers as well as responding to data queries. The proposed approach is implemented and evaluated in Roehrs et al. (2019) based on performance requirements such as response time and resources usage (memory, CPU, IO etc). However, the system is not immune against byzantine nodes as it relies on kafka which only provides crash fault tolerance. Moreover, the

paper mainly focuses on unifying data storage and querying. It does not provide further details on how data sharing and permission management is achieved and little attention is given to security and privacy aspect.

Xia et al. (2017) proposed a private Blockchain based data sharing scheme that consists of three layers: user layer, system management layer and storage layer. Users have to get registered via an issuer service and get private keys from a verifier service in order to participate in the network. However, the use of issuer and verifier service makes this system very much centralized.

A smart mobile Application known as health Data Gateway, was designed by Yue et al. (2016) with the aim of helping patients to own and manage access permissions to their data. The proposed system is a combination of traditional database and a gateway which is based on a firewall concept. The system is made up of three layers: storage layer, data management layer and Data usage layer. A private Blockchain cloud serves as the storage layer to provide confidentiality and integrity of data. Data is stored on the Blockchain cloud in an encrypted form and is shared with care providers and third parties according to user's permission. Health Data Gateway is one of few studies which differentiated between raw data access permission, aggregate query and time expiration for access privilege. It limits data sharing with external parties to only aggregate computation results via multi party computation rather than allowing them to view raw patient's data like care providers. But, the authors did not provide a detailed implementation and experimental evaluations.

The system proposed by Zheng et al. (2018) integrates key keeper applications and Blockchain to secure data sharing stored in an encrypted and compressed format on a cloud. The Blockchain serves as a permission validation platform while the key keeper applications hold the symmetric key to be used by the data owner for encryption and by the requester for decryption. The data owner distributes the shares of the symmetric key used for the encryption to multiple key keepers and the requester has to collect the shares from multiple key keepers via an authenticated communication channel to be able to decrypt the data.

The study in Ichikawa et al. (2017) developed a Blockchain based medical health application that enables patients with insomnia to get cognitive behavioral therapy. Users add their daily health data using their smart phone and doctors conduct therapy based on the collected data. A prototype was implemented based on a Hyperledger fabric Blockchain that consists of four peers and one membership service. An experiment was done to test the robustness of the system with respect to data integrity when one of the peers is down. A medical health record sharing system that employs permissioned Blockchain for access control management and an off-chain cloud service to store patients' data is proposed by Dubovitskaya et al. (2017) to achieve data security and privacy. Patient's data is encrypted using the symmetric key of the user before being uploaded to the cloud. The authors validated their work by implementing the proposed system based on a Hyperledger fabric with a network size of four nodes and a single Membership service.

The authors in Xia et al. (2017) proposed a Blockchain based system that provides data access control, provenance and auditing for sharing medical data between multiple cloud service providers. The performance of the system is evaluated using Apache Jmeter tool. The latency of the system is measured against the number of users/number of service requests to the cloud. However, this model is designed for data sharing between cloud service providers and does not consider interaction between patients, care providers and researchers.

The performance evaluation of Blockchain based healthcare systems have also been studied by other researcher (Ismail and Materwala, 2020; Thwin and Vasupongayya, 2020). Ismail and Materwala (2020) compared the performance of the traditional healthcare systems with a Blockchain based counterpart based on two performance metrics: execution time and amount of data transferred with respect to the number of records and hospitals. However, the approach uses communication time to calculate execution time and does not consider processing time.

Healthchain (Xu et al., 2019) introduced a Blockchain based privacy aware scheme for sharing electronic health record collected from internet of things devices. The system stores data on a distributed file system known as interplanetary file system (IPFS) to achieve high integrity and resiliency and to reduce the impact of a single point of failure. The proposed method was validated via simulation.

Table 1 provides a comparison of the most important related work based on various features. The majority of previous work relies on Hyperledger Fabric and Ethereum as underlying blockchain platforms. The most dominantly used consensus algorithms are RBFT and PBFT. Few studies also exist which rely on PoW and PoA consensus algorithms. Different categories of patient's data have been considered by existing research, including EHR, PHR and EMR. Two factor-authentication where both the health care provider and the user authenticate the data before it is added to the blockchain has also been considered by some studies. When it comes to data storage, three different data storage schemes are employed by existing research: off-chain, on chain and providers database. In most cases, data is stored off-chain such as in the cloud. In some cases, data is stored on chain (i.e. on the blockchain itself). There exist also some solutions which store the data at the same location where it is generated (on the provider's database). Decentralized access involves the ability of the system to provide access to data in a decentralized way which requires data to be replicated across various locations. Most existing work which provides decentralized access to data is based on on-chain storage and replication. With regards to permission, most existing research does not allow the setting of fine-grained access permission to patient's data. Moreover, they do not consider data access expiration and revocation. Once a user is granted permission to data, he/she can access it indefinitely. On the other hand, none of the previous studies provided a solution as to how care providers can have access to patient's data during emergency situations where the user is unable to give permission to doctors (See Table 2).

In summary, one of the major limitations in existing Blockchains based healthcare systems is that health record data is either stored in a centralized location (i.e. off-chain cloud, provider database, etc.), which brings about the same security risk as centralized systems or on-chain which raises performance and privacy problems. The other big limitation is that little or no attention is given to the performance and scalability challenges of Blockchains systems. To the best of our knowledge, there is no study that considers the performance and scalability challenges in the context of Blockchains based healthcare systems. Nevertheless, there are various ongoing efforts to enhance the performance and scalability of Blockchains systems though they are not directly related to healthcare systems (Amiri et al., 2019; Gupta et al., 2020; Amiri et al., 2019; Gorenflo et al., 2020). The majority of these studies focus on improving the consensus level. However, they are not yet available as full Blockchain platforms and they do not fully support smart contracts, which is the core for healthcare based applications. Caper (Amiri et al., 2019) introduced a cross-application permissioned Blockchain platform that partitions the network into clusters where each cluster represents a single application. The system employs two levels of consensus. Each application runs its own local consensus for internal transactions and a global consensus is performed between applications for cross application transactions. By doing so, the proposed method allows processing

Table 1
Summary of Related Work.

Paper	Blockchain Type	Consensus algorithm	Two factor Authentication	Data Type	Data Location	Decentralized Access	Fine Grained Permission	Permission Expiry/ Revoking	Performance Eval	Emergency Data Access
Azaria et al. (2016)	Ethereum	PoW	✓	EHR	Providers Database	X	X	X	X	X
Rouhani et al. (2019)	Hyperledger	RBFT	✓	PHR	Off-Chain Server	X	X	X	X	X
Al-Karaki et al. (2019)	Hyperledger	RBFT	X	EHR	Off-Chain Cloud	X	X	X	X	X
Tanwar et al. (2020)	Hyperledger	RBFT	X	PHR	On Chain	✓	X	X	✓	X
Roehrs et al. (2017)	Apache kafka	Kafka	X	PHR	On Chain	✓	X	X	✓	X
Yue et al. (2016)	-	-	X	PHR	On Chain	✓	✓	✓	X	X
Ichikawa et al. (2017)	Hyperledger	PBFT	X	PHR	On Chain	X	X	X	X	X
Dubovitskaya et al. (2017)	Hyperledger	PBFT	X	EHR	On Chain and Providers Database	✓	✓	✓	X	X
Daraghmi et al. (2019)	Ethereum	PoA	X	EMR	Providers Database	X	✓	✓	✓	X
Xu et al. (2019)	-	-	X	EHR	Off Chain	✓	X	X	✓	X

Table 2
Table:Notations.

Notation	Description
PR	Private Key
PR^s	System Private key
PU	Public key
PU^s	System Public key
K	Symmetric key
$H(m)$	Hash of message M
$E_k(M)$	Encryption of M with k
$SIG_{PR_l}M$	Signature of M by PR_l

internal transactions in parallel which significantly improves transaction throughput. However, the proposed system is specifically designed for supply chain applications. SharPer (Amiri et al., 2019) proposed a similar system to Caper with the additional property that it supports cross-application transactions that access only a subset of the applications. ResilientDB (Gupta et al., 2020) partitions the network into clusters where each cluster runs a local consensus on a disjoint set of transactions followed by a global sharing step where the clusters exchange the result of local consensus with each other. FastFabric (Gorenflo et al., 2020) introduced various optimization on top of HyperledgerFabric (Androulaki et al., 2018) to enhance transaction throughput.

3. Architecture of the proposed system

In this section, we present the proposed Blockchain based architecture as shown in Fig. 1 which consists of three layers: Decentralized storage layer, Blockchain layer and User Layer. The decentralized storage layer consists of an off-chain distributed file system that stores users' encrypted data and indexed using associated hashes. The Blockchain layer serves two purposes: it provides a public platform that maintains metadata and ownership information about the files stored in the decentralized file storage and also provides permission management services for data sharing between untrusting entities. The user layer involves external users who interact with the Blockchain via a decentralized application.

3.1. Blockchain layer

The proposed system is based on a permissioned Blockchain where a set of pre-specified nodes take the responsibility of mining. These nodes are collectively trusted by the rest of the network for validating transactions and creating new blocks. In our case, the trusted authorities consist of healthcare providers such as hospitals, insurance companies and regulators. The trusted authorities perform various functions including adding data to the decentralized file system, uploading the corresponding transaction to the Blockchain, validating various transaction received from external users such as permission request and permission granting.

3.1.1. Consensus

The proposed system is based on the IBFT consensus algorithm. IBFT is a voting based consensus algorithm where the majority of the nodes have to vote in order to create a valid block. In a network that consists of n validators, at least $2f + 1$ of the nodes should vote for the block where $f < n/3$ is the number of malicious nodes. IBFT consensus consists of three communication steps: The first step is a pre-processing step where the leader node prepares and sends a block proposal to all other validators. This step has a complexity of $O(n)$. In the second step, each node exchanges a prepared message with every other node. The third step is called a commit step where nodes exchange a commit message with each other. Both the second and the third steps have $O(n^2)$ complexity.

3.1.2. Smart contracts

The Blockchain layer consists of three types of smart contracts: registry contract, data contract and permission contract.

- **Registry Contract:**To protect the system from malicious users who try to add fake data or misuse data, all users are anonymously registered on the registry contract. The data consists of user public keys and their roles such as patients, care providers, researchers and regulators. Miners check the legitimacy of the user when validating transactions sent to the data contract and the permission contract.

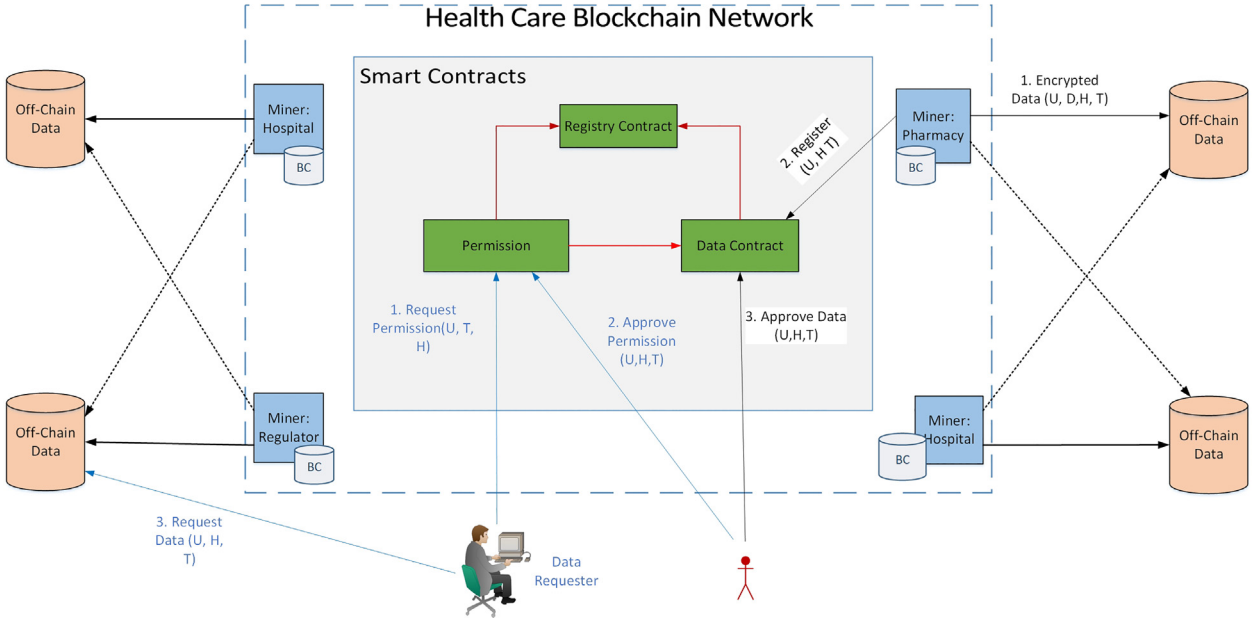


Fig. 1. Blockchain Based Healthcare Architecture.

- **Data Contract:**The data contract stores a list of records that indicates the mapping between users and their data. Each data row in the list consists of the public key of the data owner, symmetric key encrypted using the public key, the type of data and the hash of the data that points to the raw data stored off-chain. For this purpose, the data contract provides functional interfaces to add and modify data. Data is uploaded by the care providers on behalf of the user. The user can also modify the data once it is uploaded to the Blockchain.
- **Permission Contract**A permission contract keeps a record of access permissions that show the different privileges users have on data that resides in the data contract. Each access permission has three tuples: the public key of the permission granter, the public key of the permission requester, a symmetric key encrypted using the requester's public key and the hash of the data. This contract exposes functions that allow participants to request permissions, grant permissions and modify permissions.

3.1.3. User layer

Each user installs a decentralized application that allows interaction with the Blockchain and the distributed file system. Each user is equipped with a public private key pair used for signing transaction and also used as a unique address to identify the user on the Blockchain.

3.1.4. Decentralized storage layer

The decentralized storage layer is based on a peer-to-peer distributed file system known as InterPlanetary File System (IPFS). IPFS is a trustless and decentralized system similar to Blockchain but for file storage and sharing. It does not have a single point of failure and does not rely on a trusted third party. The design of IPFS integrates various existing concepts such as Distributed Hash Table (DHT) and Bittorent.

4. Proposed system operation details

In this section, we discuss the detail operations of the proposed decentralized electronic healthcare system.

4.1. Security assumptions

- **Care Providers/Miners:** Miners are not malicious. They are trusted for the data they generate. However, they are curious to know about users' data created by other care providers
- **User:** Could try to breach the confidentiality of data or modify or delete others users' data
- **Researchers:** These are external entities who could act maliciously.
- **DFS:** The distributed file system is secure and data cannot be modified or deleted by unauthorized users.

4.2. System initialization

System initialization is needed to ensure proper operations which include deploying the three smart contracts deployed on the Blockchain. System entities each generates a public private key pair $\{PR, PU\}$ and register on the Blockchain anonymously using their public keys by sending a registration transaction signed by their private key to the Registry Contract. To ensure privacy, the set of authorities utilize a threshold signature to sign users' data submitted to the Blockchain. This helps hiding relationships between care providers and users which might expose the users to privacy linking attacks. To achieve this, a single public key is used to represent the set of miners in the system and shares of the corresponding private key is secretly shared among all of them. Therefore, a system public key PU^s and a system private PR^s key is generated at the beginning and the shares of the private key is distributed to each miner $\{PR_1^s, PR_2^s \ldots .PR_N^s\}$ where N is the number of miners. Note that the miners also have an additional key pair similar to ordinary users other than the system key. The system key is used to protect the privacy of the users during data recording on the Blockchain whereas their individual key is used to perform their own private transactions such as requesting user data from another care provider.

4.3. Data format

Users' data is represented in the system as a set consisting of quadruples $\{PU, HI, DT, E_K\{RD\}\}$ where PU is the public key of the

data owner, HI is a hash index, DT is the data type and $E_K\{RD\}$ is the encrypted form of the raw data RD.

4.4. Care providers adding electronic health record

Data can be added to the system by several care providers/miners or users themselves. Suppose miner i wants to upload data that belongs to user j. There are two types of actions the miner has to perform. Storing the encrypted data on the DFS and sending the equivalent transaction to the Blockchain. Let SD and BT represent the stored data and Blockchain transaction for uploading metadata respectively. Two types of actions are performed by the care provider to add data to the system as explained below.

a. Uploading Data to the DFS:

- **Encrypt the raw data**: Miner i generates new symmetric key k and encrypts RD with key k, $E_k\{RD\}$
- **Generate Hash Index**: Miner i generates a hash for the encrypted data.
  $HI = H(E_k\{RD\})$
- **Store Data on DFS**: The message sent to DFS for data storage is:
  SD = $PU_j||DT||HI||E_k\{RD\}$

b. Sending the Metadata to Blockchain: The miner then prepares the corresponding Blockchain transaction consisting of the metadata as follows:

- **Encrypt key k**: Miner i encrypts key k with the public key of user j $E_{PU_j}\{k\}$
- **Prepare Blockchain Transaction BT** The body of the Blockchain transaction is: BT = $PU_j||DT||HI||E_{PU_j}\{k\}$.
  The symmetric key k is encrypted using the user's public key to allow the user to decrypt and get the key later. Thus, only the original data creator and the user will have access to the symmetric key.
- **Threshold sign BT**: Miner i signs BT using the share of his private key $SIG_{PR_i^s}\{BT\}$
- **Multi cast BT and collect Partial signatures**: Miner i sends BT to other miners for their signature and collect a minimum of t partial signatures out of N miners, $SIG_{PR_x^s}\{BT\}$ where $x = 1 \ldots t$. Note that this has two benefits. First, only miner i has access to the raw data. All other miners only see the hashed and encrypted forms. Second, external parties will not be able to know which care provider uploaded the data on behalf of user j.
- **Combine Partial Signatures**: Combine the Partial signatures collected from other miners to generate a signature of PR^s on BT, $SIG_{PR^s}\{BT\}$
- **Send Transaction to Blockchain**: The final transaction to be sent to the Blockchain is $SIG_{PR^s}\{BT\}||BT$

Algorithm 1 shows the pseudocode explaining the steps followed by care providers when adding data to the Blockchain and

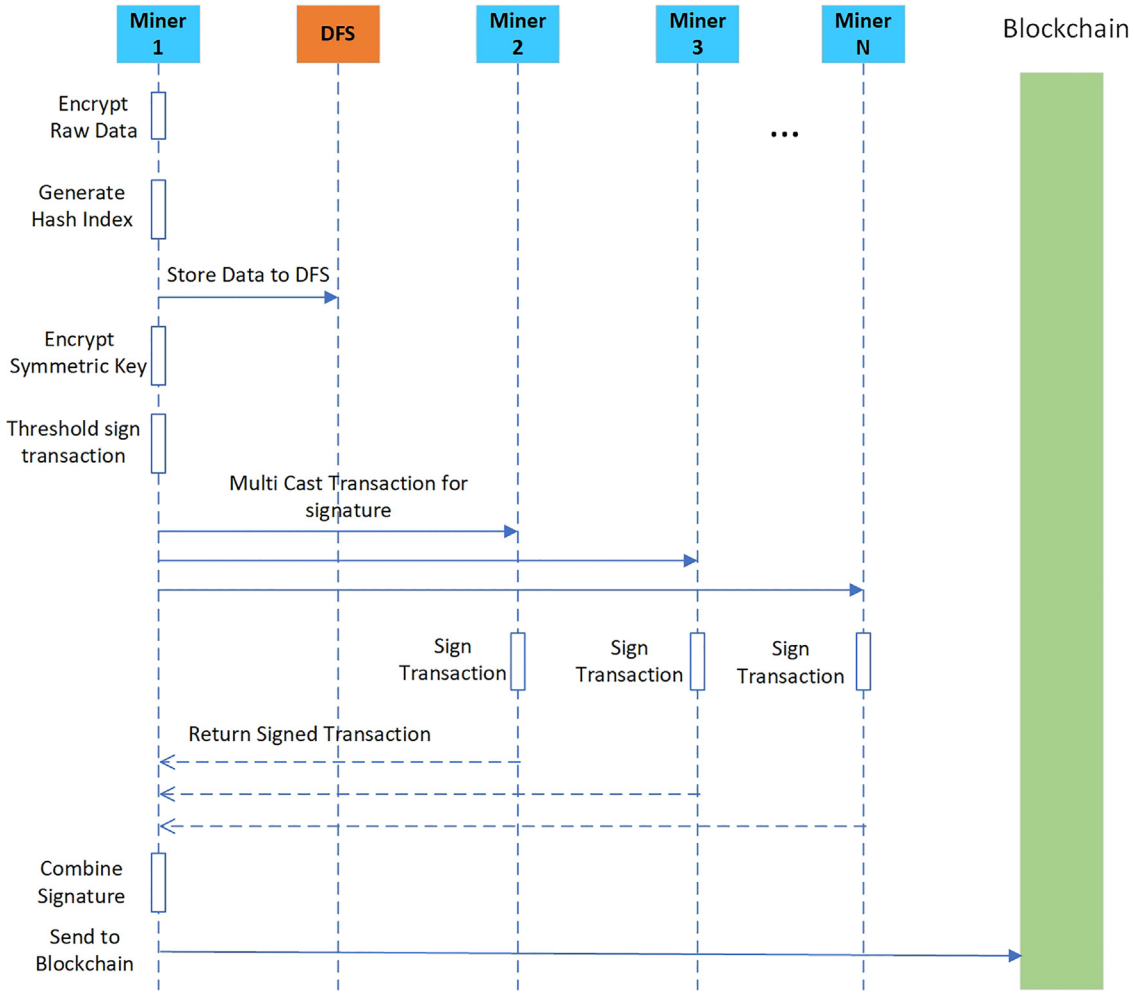


Fig. 2. The steps of Adding data to the DFS and Blockchain.

DSF. The pseudocode particularly shows when miner i adds data for user j.

Algorithm 1: Care Providers Adding Electronic Health Record Data

Input: Raw Data RD, System Private Key PR^s, Secret Share of Miner x on System Private key PR^s_x
Symmetric Key K, Public Key of User j PU_j, Number of Miners N, Threshold t, Data Type DT
Output: Stored Data SD, Signed Blockchain Transaction BT
Stage I: Uploading Data to the DFS 1: Encrypt Raw Data $\Rightarrow$ $E_k\{RD\}$
2: Generate Hash Index $HI = H(E_k\{RD\})$
3: Store Data on DFS, SD = $PU_j||DT||HI||E_k\{RD\}$
Stage II: Sending the Metadata to Blockchain
4: Encrypt Symmetric key using $PU_j \Rightarrow E_{PU_j}\{k\}$
5: Prepare Blockchain Transaction BT = $PU_j||DT||HI||E_{PU_j}\{k\}$
6: Threshold sign BT $\Rightarrow SIG_{PR^s_i}\{BT\}$
7: Multi cast BT to the N miners
8: Collect a minimum of t partial signatures on BT $SIG_{PR^s_x}\{BT\}$ where $x = \{1 \ldots t\}$
9: Combine Partial Signatures $\Rightarrow SIG_{PR^s}\{BT\}$
10: Send Transaction to Blockchain $SIG_{PR^s}\{BT\}||BT$

The sequence diagram in Fig. 2 shows an example scenario that explains the steps of adding data to the DFS and the Blockchain. There are N miners in total and $Miner_1$ is the data generator. $Miner_1$ encrypts the raw data using a symmetric key, generates the hash and then stores the encrypted data to the DFS. $Miner_1$ then encrypts the symmetric key using the public key of the user and prepares a Blockchain transaction that consists of the encrypted symmetric key and the hash of the data. Next, $Miner_1$ sends the corresponding Blockchain transaction to the other miners for threshold signature. Each miner threshold-signs the transaction and sends it back to $Miner_1$. $Miner_1$ then combines the threshold signature and sends the transaction to the Blockchain network.

4.5. User adding personal health record

Users can also add personal health records to the system if they wish to. The process is similar to the steps explained in Section 4.4. However in this case, there is no need for multi signature signing. The Blockchain transaction is signed by the user's private key. Moreover, the encrypted key is not included in the Blockchain transaction as in the case of care providers. The process of adding data involves the same two actions as explained earlier.

a. Uploading Data to the DFS:
This step consists of the same three steps explained previously in Section 4.4.

b. Sending the metadata to Blockchain: Sending the Blockchain transaction is accomplished in two steps:

- **Create Blockchain transaction**: The content of the Blockchain transaction is BT = $PU_j||DT||HI$.
- **Sign and send the Transaction**: The transaction to be sent to the Blockchain is signed as $SIG_{PR_j}\{BT\}||BT$

Algorithm2 represents the pseudocode used by users when adding PHR data to the Blockchain and DSF. The pseudocode explains the scenario where user j adds data.

Algorithm 2: Users Adding Personal Health Record Data

Input: Raw Data RD, Private Key of user j PR_j, Public Key of User j PU_j, Symmetric Key K, Data Type DT
Output: Stored Data SD, Signed Blockchain Transaction BT
Stage I: Uploading Data to the DFS 1: Encrypt Raw Data $\Rightarrow$ $E_k\{RD\}$
2: Generate Hash Index $HI = H(E_k\{RD\})$
3: Store Data on DFS, SD = $PU_j||DT||HI||E_k\{RD\}$
Stage II: Sending the Metadata to Blockchain
4: Create Blockchain transaction BT = $PU_j||DT||HI$
5: Sign and send the Transaction $\Rightarrow SIG_{PR_j}\{BT\}||BT$

4.6. Querying data

When a new transaction consisting of the metadata is sent to the Blockchain, a notification is sent to the concerned user. Once the user receives the notification, he downloads the metadata from the Data Contract including the hash index and the encrypted symmetric key, k. Next, the symmetric key is decrypted using the user's private key. The encrypted data is then retrieved by sending a query that consists of the hash index to one of the nearby DFS nodes. Once the user receives the encrypted data, the raw data is decrypted using the symmetric key.

4.7. Requesting permission

Once the metadata is available on the Blockchain, other users or care providers can see who has what data. Hence, anyone interested in the data can submit a permission request to the Blockchain network by sending a transaction that invokes the Permission contract.There are two types of data sharing: Incentive based and non-incentive based sharing. Users share their data with care providers such as doctors and nurses for free. However, users might want to get incentives to share their data with third parties such as researchers or marketing organizations. For this reason, the Permission contract provides different functional interfaces for incentive-based and non-incentive based data access. The permission request transaction (PRT) takes the following format:
$PRT = SIG_{PR_i}\{HI\}||HI$where i is the user requesting the data.

4.8. Granting permission

When a permission request is sent to the Permission contract to access a specific data, the data owner receives a notification and can either approve or reject the request. If the request is approved, a transaction is sent consisting of: the HI of the requested data, the public key of the requester and the symmetric key used to decrypt the requested data encrypted with the public key of the requester. The permission grant transaction (PGT) takes the following format:
$PGT = SIG_{PR_j}\{HI, PU_i, E_{PU_i}\{k\}\}||\{HI, PU_i, E_{PU_i}\{k\}\}$where i is the permission requester and j is the data owner approving the request. Once the permission is approved, the user downloads the encrypted key and decrypts the symmetric key. The data then can be retrieved from a nearby DFS node by using the HI and decrypted using the symmetric key.

5. Implementation

Our proposed system is implemented based on an Enterprise Ethereum Blockchain platform known as Hyperledger Besu. It is a permissioned Blockchain that allows specifying a set of trusted

nodes to act as miners. We use IPFS (Benet, 2014) for decentralized data storage. A Blockchains client application utilizes a JavaScript object notation remote procedure call (JSON-RPC) protocol to connect with Blockchain nodes. In our case, we use a Java based Web3 API known as Web3j that provides JSON-RPC interface to communicate with the Blockchain. Fig. 3 represents the implementation of a care provider node. It consists of a main engine, a local database, an IPFS client, a threshold manager and an Ethereum client. The main engine is the starting point which coordinates all other tasks on the node. The local database stores all data created locally. The IPFS client allows storing data to the DFS. The purpose of the threshold manager is to facilitate the process of threshold signature between nodes via multi-cast communication. The various software components making up the care provider node are developed in Java. Patients do not host a full node. They use their decentralized App (DApp) to communicate with a remote Ethereum client and IPFS clients. The DApp allows the user to submit transactions and read data.

5.1. Smart contracts

Fig. 4 shows the design of the three smart contracts and the interaction between them. The Registry contract provides two functional interfaces: *registerUser()* and *checkUser()*. The *registerUser()* function allows users, care providers and researchers to register based on their public keys while *checkUser()* is invoked by the other two smart contracts to check the validity of the user at the time of adding data and permission approval. The Metadata Contract invokes four functions. The *addEHRData()* and *addPHRData()* are invoked when adding EHR and PHR data respectively. The *notifyUser()* functions sends notification when metadata belonging to the user is added. Finally, the *searchData()* enables other participants to explore what data is available. The Permission contract accepts permission request and approval transactions. There are two types of permission requests: incentive based (INBased) permission request and non incentive based (NINBased) permission request. Accordingly, the Permission contract exposes four functional interfaces: *requestINBasedPermission()*, *requestNINBasedPermission()*, *approveNINBasedPermission()* and *approveINBasedPermission()* for permission request and approval. Our smart contracts are developed in the solidity programming language.

6. Evaluation

We conducted several experiments to evaluate the performance of the proposed system. We compared the proposed Blockchain based approach with existing PoW based solutions such as MedRec (Azaria et al., 2016) and also with traditional database systems. PoW is a probabilistic computational intensive consensus algorithm that involves solving a cryptographic puzzle to prove that a block is valid. PoW is considered secure as long as 51% of the nodes hold the total computation power. The following sections describe the implementation, experimental environment and configurations used in the experiment.

6.1. Evaluation metrics

This section describes the performance evaluation metrics used in the experiments.

- **Transaction Latency:** There are two types of transactions: write transaction and read transaction. Unlike a read transaction, a write transaction is a transaction that changes the state of the Blockchain such as Adding EHR data to the Blockchain. For a write transaction, the latency is defied as the time taken for a transaction to be committed on the Blockchain starting from the time when it was submitted by the user. A transaction is considered committed once included in a block. Each block carries a timestamp indicating the time when the block was created. Thus, we consider the timestamp of the block as the committed time. IBFT consensus provides immediate finality. Thus, a transaction is considered final the first time it appears on a block. In other words, the timestamp of the first block is taken as the committed time of the transaction. However, for PoW which is a probabilistic based consensus approach, a transaction is considered committed after a certain number of blocks have been added on top of the block containing the transaction.

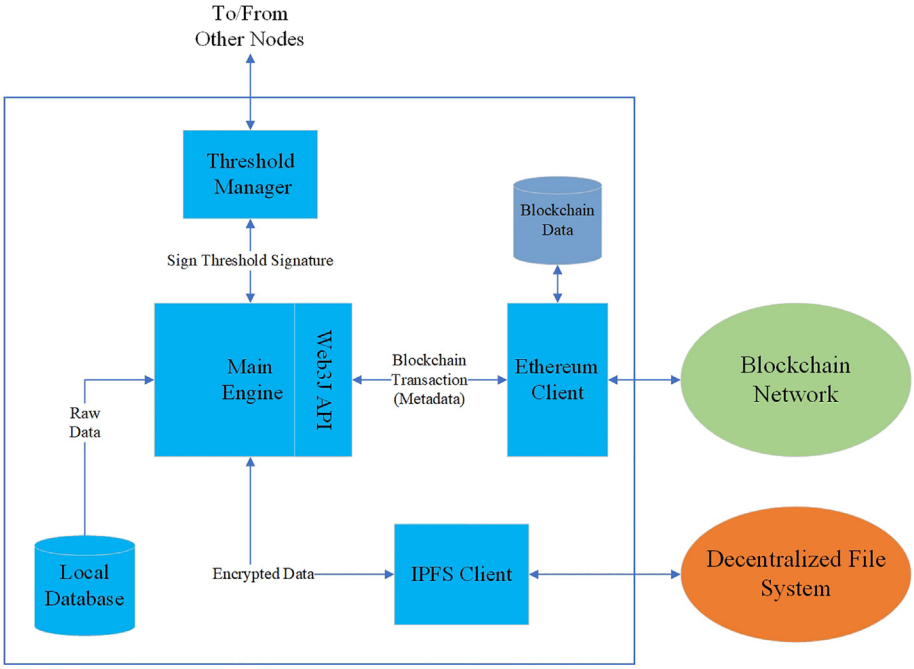


Fig. 3. Components of Care Provider Node.

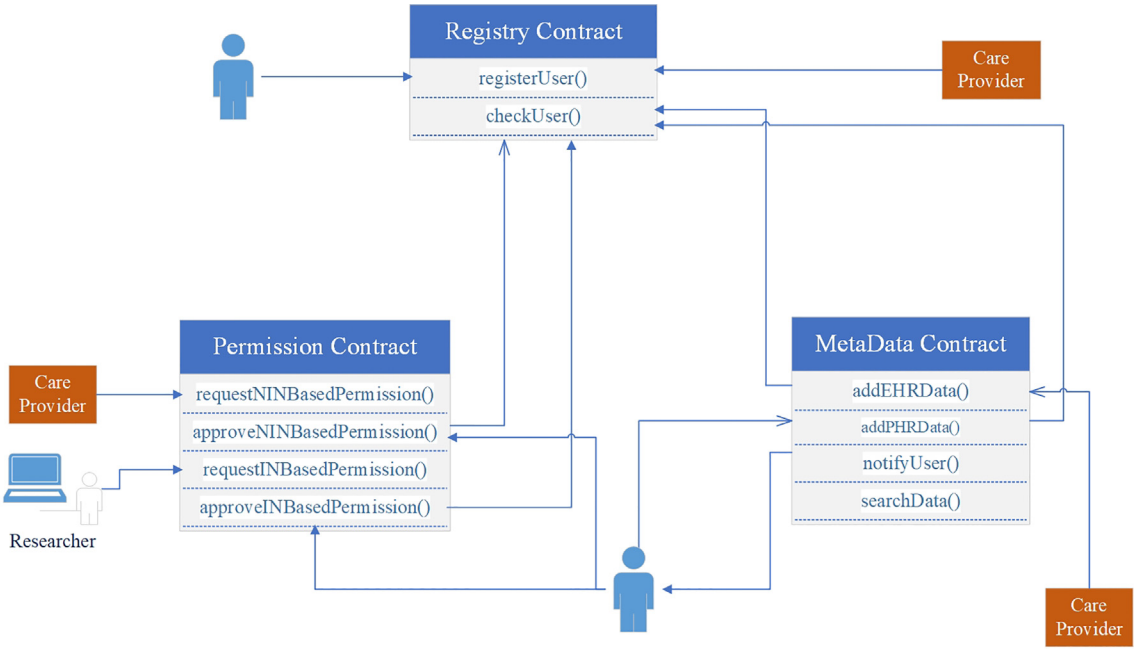


Fig. 4. Smart Contracts and Their Interaction.

The recommended number of such blocks is 6 (Buterin, 2020). Thus, we take the timestamp of the 6^{th} block as the committed time. The latency for a read transaction such as querying EHR data from a Blockchain is defined as the difference between the submission time and the response time where the response time is the time when the query result is returned to the user.

- **Transaction Throughput:** is defined as the number of successful transactions divided by the time interval between the last committed (response) time and the first submission time.
- **Scalability:** This metrics indicates how the latency and throughput change as the number of transactions or nodes changes.
- **Fail Rate:** A transaction is considered a failed one if the transaction is not committed or a response is not received within a certain time period. A waiting time of 15 min is used which is the default in an Ethereum Blockchain.

6.2. Experimental Environment and Dataset

The experiments were conducted on Ubuntu virtual machine that has 120 GB RAM and 16 CPU cores each with 2 GHz speed. To look at the scalability of the system, the experiments were performed with different number of nodes: 10, 30 and 50. Each node runs an instance of the Hyperledger Besu Blockchain client software. We implemented the user client application using Java programming where each user is represented by a separate application process and each user is assigned a pair of public–private keys. We used the Covid-19 dataset published on this link (Beoutbreakprepared, 2020). The data consists of around 1 million records related to Covid-19 patients from multiple countries. It describes Covid-19 patients based on various features including the country, province, city, location, travel history, date of confirmation and outcome (i.e whether the patient passed away, recovered, etc.).

6.3. Parameter configuration

We configured different network parameters that can affect the performance such as Block gas limit, Block period seconds and difficulty. Block gas limit determines the maximum computational steps (gas) allowed for a block. It affects the number and weight of transactions that can be included in a block. This limit is required for security reasons i.e. to prevent attackers from sending a transaction that causes an infinite loop. Block period seconds determines how often blocks can be created in IBFT. The difficulty controls the level of difficulty in solving the cryptographic puzzle in PoW mining. The gas limit is set to the maximum possible value of ("0x1fffffffffffff") as the proposed system is a permissioned Blockchain. The Block period was set to 1 s and the difficulty level was set to 1000 to get a maximum performance.We then prepared a test schedule that consisted of 10 rounds where we increased the number of transactions from 1000 to 10,000 in an increment of 1000. We ran each test 5 times and we took the average of the 5 tests.

7. Experimental results and discussion

In this section, experimental results are presented along with detail discussions. The section is divided into four subsections. The first subsection is related to peak performance and the second and third subsections are dedicated to scalability. The last subsection compares the performance of the decentralized file system with a traditional database system.

7.1. Peak performance

We measured the peak performance of the proposed system while keeping the network size constant at 10 nodes. The peak latency and peak throughput are shown in Figs. 5 and 6. As can be seen from the Figures, the proposed system which is based on IBFT, shows better performance as compared to PoW in all cases except for query EHR Data latency which is a read transaction that does not require a consensus process. In case of query EHR Data, both systems show similar performance. Moreover, the latency for a query transaction is very low compared to an add EHR Data transaction which involves consensus. The proposed system has a peak latency of 1507 ms which is 14x lower than that of PoW (21275 ms) and provides 2.5x throughput (62 transactions per

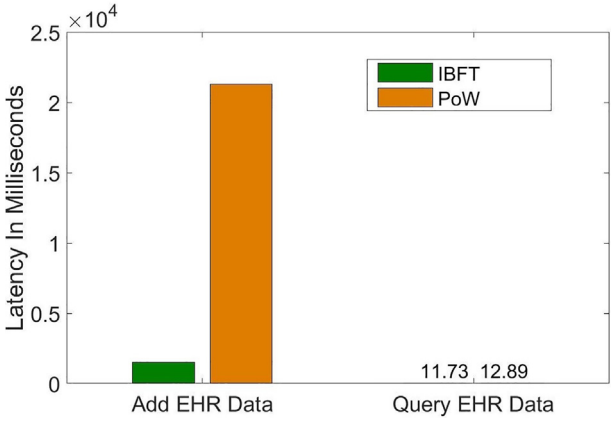


Fig. 5. Peak Latency.

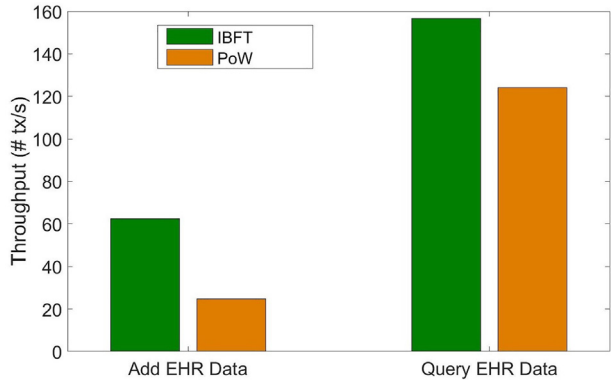


Fig. 6. Peak Throughput.

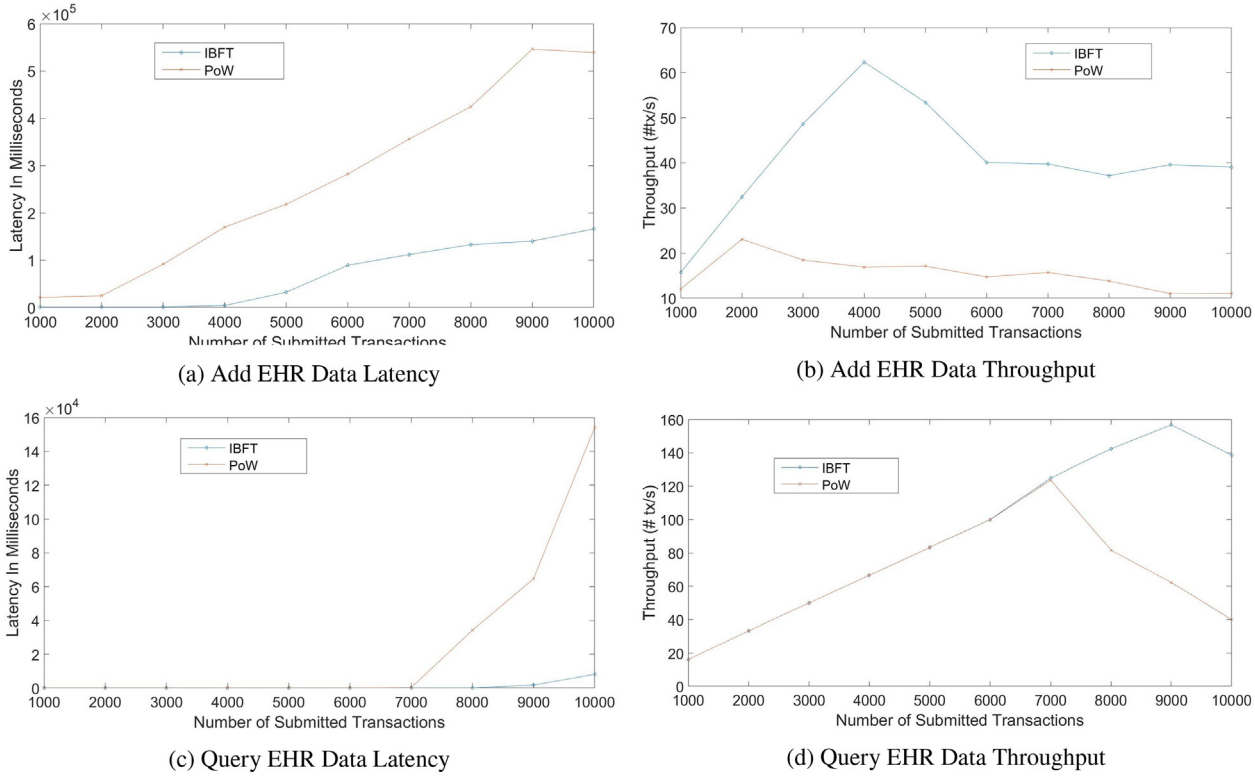
second) compared to PoW (24.7 transactions per second). No fail rate is recorded for both systems.

7.2. Latency/throughput vs number of transactions

We monitored the change in latency and throughput by varying the number of transactions from 1000 to 10,000 transactions per minute. The latency of both systems increases monotonically as the number of transactions increases. AS for throughput, it increases for sometime until it reaches a maximum limit/saturation point before it starts to degrade. The latency of the proposed system stays always lower than that of PoW while the throughput remains higher than that of PoW as shown in Figs. 7d. For Query EHR Data latency, the proposed system showed an increment of only 695× (from 11.73 to 8154 ms) as compared to PoW which rised by 12299× (12.89 to 158535 ms). Moreover, the proposed system's throughput only degraded by 1.6× and 1.14× for Add EHR Data and Query EHR Data respectively while the PoW based system throughput degraded by a factor of 2.25× and 3.68× respectively. However, the PoW based system tends to be more scalable in the case of Add EHR Data latency. Add EHR Data latency for the proposed system increased by 110× (from 1506.88 to 166324 ms) while that of PoW increased by 25× (from 21275 to 539148 ms). The lowest latency is recorded for the case of 1000 transactions per minute for both systems while the highest throughput occurred around 2000 transactions per minute for PoW and around 4000 transactions per minute for the proposed system (See Fig. 8).

7.3. Latency/throughput vs number of nodes

We repeated the same experiment above for a network size of 30 nodes. The results indicate that the proposed system still performs better compared to PoW in all cases except one as can be



(a) Add EHR Data Latency

(b) Add EHR Data Throughput

(c) Query EHR Data Latency

(d) Query EHR Data Throughput

Fig. 7. Latency/Throughput Vs Number Of Transactions at a network size of 10 Nodes.

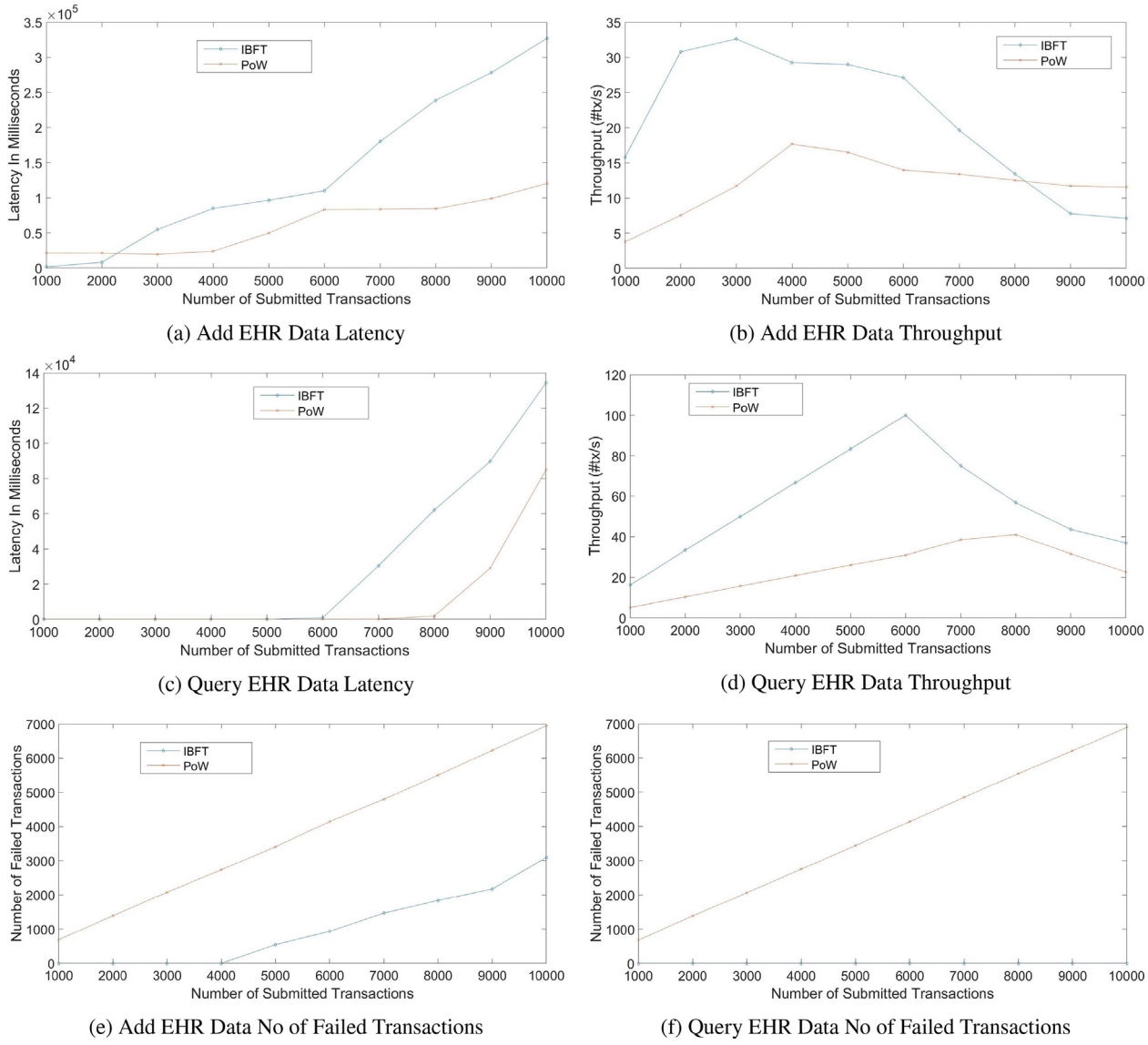


Fig. 8. Latency/Throughput Vs Number Of Transactions at a network size of 30 Nodes.

seen from Figs. 9f. In the case of Add EHR latency, PoW shows better latency. However, this is at the expense of an increased failure rate. As can be seen from Fig. 9e, PoW has a failure rate with more than half of the transactions failing for Add EHR data. Moreover, unlike PoW, the proposed system does not have any failed transactions for query EHR data. In addition, the average performance of each system was analyzed as the number of nodes was increased from 10 to 30. The summary of average performance for different network sizes is shown in Table 3. As can be seen, the number of nodes has a high impact on the performance of the system with respect to both latency and throughput. Moreover, transactions start to fail as the number of nodes is increased to 30 due to congestion which was not the case when only 10 nodes were used. Particularly, the PoW based system shows a higher number of failed transactions.

7.4. Decentralized file system vs traditional database

In this section, the performance of the proposed decentralized file system is compared with the traditional database. The proposed distributed file system was implemented based on IPFS with a network size of 10 nodes (Benet, 2014) and the traditional database is implemented using a MySql database. We tested the latency and throughput of the two systems by generating a number of transactions that vary from 1000 to 10,000 transactions per minute. A transaction involves storing or querying a single row of EHR record data to/from the respective databases. For IPFS, each row of a record is represented in a JavaScript Object Notation (JSON) file. The experimental results show that the proposed system performs better with respect to querying EHR Data while the traditional database has better performance for adding EHR data to the database as presented in Figs. 9d. For Query EHR data, the proposed system has a peak latency of 1 ms while the traditional database has a peak latency of 10 ms which is 10x times higher than that of the proposed system. The decentralized file systems provides better latency as the data can be accessed from any node in the network. As for query throughput, both systems have a maximum throughput of 166 transactions per seconds. For add EHR Data, the two systems show similar latency and throughput performance up to 4000 transactions per minute. However, the traditional database provides 4x and 2x latency and throughput respectively when compared with the decentralized file system as the number of transactions is increased further. The traditional database has a peak latency of 12.7 ms and provides a throughput

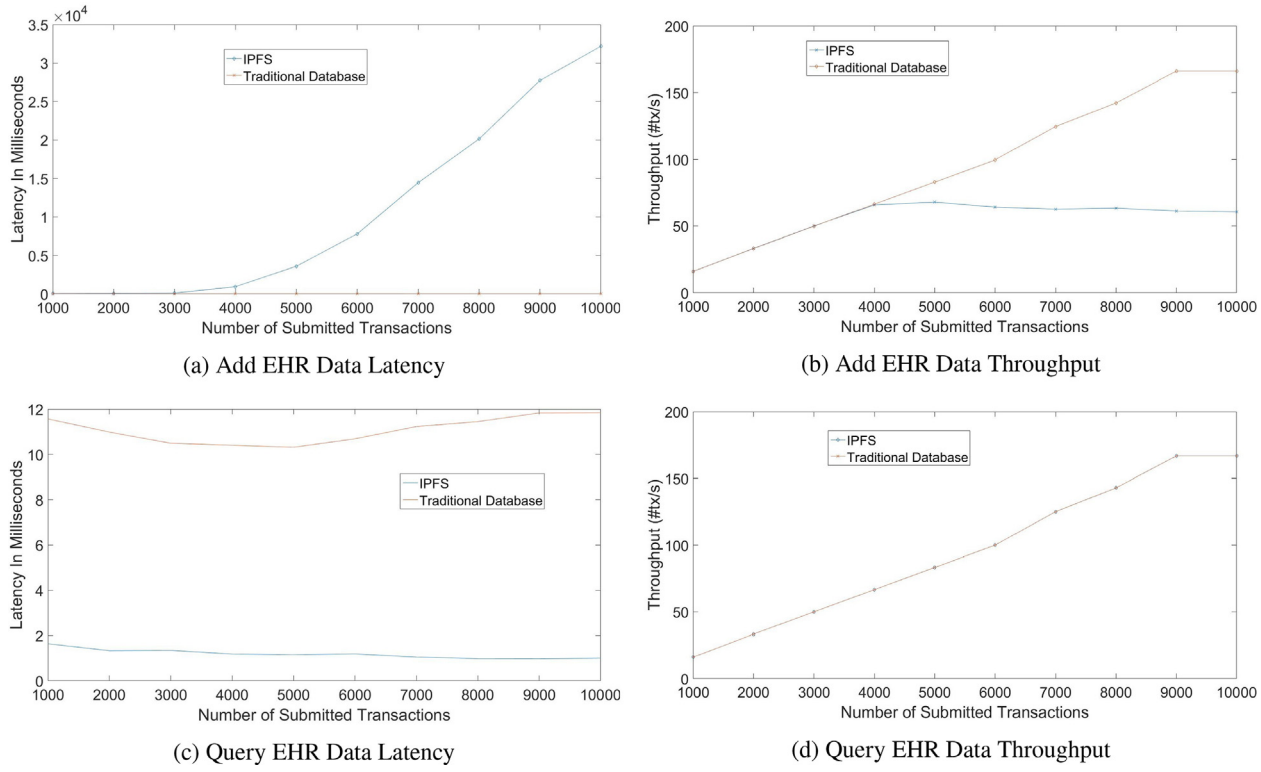


Fig. 9. Comparison between Decentralized file system with 10 nodes and traditional database.

Table 3
Summary Of Average Performance for different network sizes AEDL: Add EHR Data Latency, AEDT: Add EHR Data throughput, QEDL: Query EHR Data latency, QEDT: Query EHR Data throughput.

Type	Proposed System		PoW	
	10	30	10	30
AEDL	68248	137970	267464	60598
AEDT	40	21	15	12
QEDL	1013	31746	24787	11593
QEDT	91	56	66	24

of 165.96 transactions per second. On the other hand, the peak latency and throughput for IPFS is 55 ms and 70.99 transactions per second respectively. The decentralized file system uses hashing to preserve the integrity and security of the data which justifies the performance degradation noted at higher transaction rates.

8. Conclusions

In this paper, a decentralized electronic health record system was proposed which provides better security and efficient sharing of electronic health record data as compared to existing systems. The proposed system integrates three types of systems: permissioned Blockchain based on an IBFT consensus algorithm, a threshold signature approach and a decentralized file system. The proposed system was implemented based on the Hyperledger Besu Blockchain platform and the Interplanetary file system. Several experiments were performed and the performance results of the proposed system were compared with existing Blockchain based solutions and traditional database systems. The experimental results show that the proposed system performs better than existing Blockchain based solutions in most cases and under different network sizes while providing better security and integrity of data.

Funding

This work is supported by reearch grant number 31R180 from the Zayed Center for Health Sciences, UAE Univeristy, UAE.

Declaration of Competing Interest

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

References

Al-Karaki, J.N., Gawanmeh, A., Ayache, M., Mashaleh, A., 2019. Dass-care: A decentralized, accessible, scalable, and secure healthcare framework using blockchain, in: 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), pp. 330–335.

Amiri, M.J., Agrawal, D., Abbadi, A.E., 2019a. Caper: A cross-application permissioned blockchain. Proc. VLDB Endow. 12, 1385–1398. https://doi.org/10.14778/3342263.3342275, 10.14778/3342263.3342275.

Amiri, M.J., Agrawal, D., Abbadi, A.E., 2019b. Sharper: Sharding permissioned blockchains over network clusters. CoRR abs/1910.00765. http://arxiv.org/abs/1910.00765, arXiv:1910.00765.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al., 2018. Hyperledger fabric: a

distributed operating system for permissioned blockchains, in: Proceedings of the thirteenth EuroSys conference, pp. 1–15.
Azaria, A., Ekblaw, A., Vieira, T., Lippman, A., 2016. Medrec: Using blockchain for medical data access and permission management, in. In: 2016 2nd International Conference on Open and Big Data (OBD), pp. 25–30.
Benet, J., 2014. IPFS - content addressed, versioned, P2P file system. CoRR abs/1407.3561. http://arxiv.org/abs/1407.3561, arXiv:1407.3561.
Beoutbreakprepared, ncov2019/latest_data at master beoutbreakprepared/ncov2019. https://github.com/beoutbreakprepared/nCoV2019/tree/master/latest_data. (Accessed on 12/13/2020).
Buterin, V., On slow and fast block times — ethereum foundation blog. https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/. (Accessed on 12/04/2020).
Buterin, V. et al., 2014. Ethereum white paper: a next generation smart contract & decentralized application platform. First version 53.
Daraghmi, E., Daraghmi, Y., Yuan, S., 2019. Medchain: a design of blockchain-based system for medical records access and permissions management. IEEE Access 7, 164595–164613.
Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F., 2017. Secure and trustable electronic medical records sharing using blockchain. In: AMIA annual symposium proceedings. American Medical Informatics Association, p. 650.
Gorenflo, C., Lee, S., Golab, L., Keshav, S., 2020. Fastfabric: Scaling hyperledger fabric to 20 000 transactions per second. International Journal of Network Management 30, e2099. https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2099, doi: 10.1002/nem.2099, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.2099. e2099 nem.2099.
Gupta, S., Rahnama, S., Hellings, J., Sadoghi, M., 2020. Resilientdb: Global scale resilient blockchain fabric. Proc. VLDB Endow. 13, 868–883. https://doi.org/10.14778/3380750.3380757, 10.14778/3380750.3380757.
IBM, Ibm report: Compromised employee accounts led to most expensive data breaches over past year - jul 29, 2020. https://newsroom.ibm.com. (Accessed on 09/02/2020).
Ichikawa, D., Kashiyama, M., Ueno, T., 2017. Tamper-resistant mobile health using blockchain technology. JMIR mHealth and uHealth 5, e111.
Ismail, L., Materwala, H., 2020. Blockchain paradigm for healthcare: performance evaluation. Symmetry 12, 1200.
Journal, H., Healthcare data breach statistics. https://www.hipaajournal.com/healthcare-data-breach-statistics/. (Accessed on 09/02/2020).
Kumar, A., Krishnamurthi, R., Nayyar, A., Sharma, K., Grover, V., Hossain, E., 2020. A novel smart healthcare design, simulation, and implementation using healthcare 4.0 processes. IEEE Access 8, 118433–118471.
Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q., 2020. A survey on the security of blockchain systems. Future Gener. Comput. Syst. 107, 841–853. https://doi.org/10.1016/j.future.2017.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17318332.
Mazlan, A.A., Mohd Daud, S., Mohd Sam, S., Abas, H., Abdul Rasid, S.Z., Yusof, M.F., 2020. Scalability challenges in healthcare blockchain system-a systematic review. IEEE Access 8, 23663–23673.
McGhin, T., Choo, K.K.R., Liu, C.Z., He, D., 2019. Blockchain in healthcare applications: research challenges and opportunities. J. Netw. Comput. Appl. 135, 62–75. https://doi.org/10.1016/j.jnca.2019.02.027. URL: http://www.sciencedirect.com/science/article/pii/S1084804519300864.
MedRec, Medrec technical report. https://medrec.media.mit.edu/images/medrec_technical_documentation.pdf. (Accessed on 09/03/2020).
openEHR, Openehr: An open domain-driven platform for developing flexible e-health systems. https://www.openehr.org/. (Accessed on 08/30/2020).
Rahman, M.S., Khalil, I., Mahawaga Arachchige, P.C., Bouras, A., Yi, X., 2019. A novel architecture for tamper proof electronic health record management system using blockchain wrapper. In: Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure. Association for Computing Machinery, New York, NY, USA, pp. 97–105. https://doi.org/10.1145/3327960.3332392.
Roehrs, A., da Costa, C.A., da Rosa Righi, R., 2017. Omniphr: a distributed architecture model to integrate personal health records. J. Biomed. Inf. 71, 70–81. https://doi.org/10.1016/j.jbi.2017.05.012. URL: http://www.sciencedirect.com/science/article/pii/S1532046417301089.
Roehrs, A., da Costa, C.A., da Rosa Righi, R., da Silva, V.F., Goldim, J.R., Schmidt, D.C., 2019. Analyzing the performance of a blockchain-based personal health record implementation. J. Biomed. Inf. 92,. https://doi.org/10.1016/j.jbi.2019.103140. URL: http://www.sciencedirect.com/science/article/pii/S1532046419300589 103140.
Roehrs, A., da Costa, C.A., da Rosa Righi, R., Rigo, S.J., Wichman, M.H., 2019. Toward a model for personal health record interoperability. IEEE J. Biomed. Health Inf. 23, 867–873.
Rouhani, S., Butterworth, L., Simmons, A.D., Humphery, D.G., Deters, R., 2019. Medichaintm: a secure decentralized medical data asset management system. CoRR abs/1901.10645. http://arxiv.org/abs/1901.10645, arXiv:1901.10645.
Tanwar, S., Parekh, K., Evans, R., 2020. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. J. Inf. Secur. Appl. 50,. https://doi.org/10.1016/j.jisa.2019.102407. URL: http://www.sciencedirect.com/science/article/pii/S2214212619306155 102407.
Thwin, T.T., Vasupongayya, S., 2020. Performance analysis of blockchain-based access control model for personal health record system with architectural modelling and simulation. Int. J. Networked Distrib. Comput.
Xia, Q., Sifah, E.B., Asamoah, K.O., Gao, J., Du, X., Guizani, M., 2017. Medshare: trustless medical data sharing among cloud service providers via blockchain. IEEE Access 5, 14757–14767.
Xia, Q., Sifah, E.B., Smahi, A., Amofa, S., Zhang, X., 2017. Bbds: Blockchain-based data sharing for electronic medical records in cloud environments. Information 8, 44.
Xu, J., Xue, K., Li, S., Tian, H., Hong, J., Hong, P., Yu, N., 2019. Healthchain: a blockchain-based privacy preserving scheme for large-scale health data. IEEE Internet Things J. 6, 8770–8781.
Yue, X., Wang, H., Jin, D., Li, M., Jiang, W., 2016. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. 40, 1–8. https://doi.org/10.1007/s10916-016-0574-6.
Zhang, P., Schmidt, D.C., White, J., Lenz, G., 2018. Chapter one - blockchain technology use cases in healthcare, in: Raj, P., Deka, G.C. (Eds.), Blockchain Technology: Platforms, Tools and Use Cases. Elsevier. volume 111 of Advances in Computers, pp. 1–41. http://www.sciencedirect.com/science/article/pii/S0065245818300196, doi: 10.1016/bs.adcom.2018.03.006.
Zheng, X., Mukkamala, R.R., Vatrapu, R., Ordieres-Mere, J., 2018. Blockchain-based personal health data sharing system using cloud storage, in. In: 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–6.