IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 30, 2019, accepted November 5, 2019, date of publication November 11, 2019,
date of current version November 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952942

MedChain: A Design of Blockchain-Based System for Medical Records Access and Permissions Management

EMAN-YASSER DARAGHMI[1], **YOUSEF-AWWAD DARAGHMI**[2], **AND SHYAN-MING YUAN**[3]
[1]Department of Applied Computing, Palestine Technical University–Kadoorie, Tulkarm, Palestine
[2]Department of Computer Systems Engineering, Palestine Technical University–Kadoorie, Tulkarm, Palestine
[3]Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Eman-Yasser Daraghmi (e.daraghmi@ptuk.edu.ps)

ABSTRACT Although the blockchain technology was first introduced through Bitcoin, extending its usage to non-financial applications, such as managing electronic medical records, is an attractive mission for recent research to balance the needs for increasing data privacy and the regular interaction among patients and health providers. Various systems that adopts the blockchain in managing medical records have been proposed. However, there is a need for more work to better characterize, understand and evaluate the employment of blockchain technology in the healthcare industry. In this paper, a design of blockchain based system, namely MedChain, for managing medical records is proposed. MedChain is designed to improve the current systems as it provides interoperable, secure, and effective access for medical records by patients, health care providers, and other third parties, while keeping the patients' privacy. MedChain employs timed-based smart contracts for governing transactions and controlling accesses to electronic medical records. It adopts advanced encryption techniques for providing further security. This work proposes a new incentive mechanism that leverages the degree of health providers regarding their efforts on maintaining medical records and creating new blocks. Extensive experiments are conducted to evaluate the MedChain performance, and results indicate the efficiency of our proposal in handling a large dataset at low latency.

INDEX TERMS Blockchain, electronic medical records, incentive mechanism, smart contracts.

I. INTRODUCTION

Over the last decade, the adoption of new technologies for the daily management of Electronic Medical Records (EMRs) have begun worldwide. The World Health Organization (WHO) have identified medical records as assets who crave innovation and whose sharing goes far beyond their primary use. They have arisen with the potential to affect the quality of individuals' life all over the world. While various technologies have significant impact on the healthcare industry, blockchain can be considered as one of the exaggerate breakthroughs in at least half a century. Essentially, a blockchain is a distributed database solution which stores a continually increasing set of data verified and confirmed by participants. Researchers believe that the blockchain technology can shape the healthcare industry in everything from protecting medical records, offering better patient packages and streamlining billing.

In a daily basis, a patient may visit more than one health-care provider for various needs, such as general practitioner, specialists, clinics, pharmacies, etc. The EMR will be stored in the provider's database who issued the record and will only be the eligible provider for editing it. This provider also will be responsible for the record's maintenance and management. Patients' with access rights could query their EMRs from different providers. Providers' with access rights could query EMRs of a common patient from other health provider when there is a need, such as consulting related EMRs for making diagnosis. These situations cause a lack of coordinated data management and exchange. In other words, medical records are fragmented and isolated, rather than cohesive. The need for multiple access to the EMRs had raised the interoperability challenges between patients and health providers which pose additional barriers to effective

The associate editor coordinating the review of this manuscript and approving it for publication was Gautam Srivastava.

data sharing. Additionally, as technology is constantly evolving, several advanced techniques are developed to violate digital privacy and security. Unfortunately, medical records are considered as major targets for information theft since they include private and sensitive information, e.g. the patients' names, identity numbers, contacts info and addresses.

Although the blockchain technology was first introduced through Bitcoin, extending its usage to non-financial applications is a mission for researchers. The industry of healthcare is one of the fields where the blockchain technology is believed to have considerable impact. The adoption of the blockchain in the healthcare field can be found in [1]–[15]. Research in this field is relatively new but increasing very quickly. Researchers propose scalable permissioned blockchain based frameworks for EMRs where patients' medical records are stored in the existing databases of providers and the blockchain can be integrated as an access control layer to allow the interaction between participants. For example, the Guardtime [16] which is a firm that runs a blockchain based platform for performing the process of validating the identities of patients for Estonia citizens, and the MedRec [7] project that is proposed to provide permissions management, authorization and data sharing between participants. Researchers also propose varying techniques for establishing secure access controls for the blockchain, such as a biometric identity system for parties' authentication [4], and a deposit-box to transfer a record from one party to another [17]. Although a number of studies is proposed to employ the blockchain for managing EMRs, there is still a need for more research to better understand, characterize and evaluate its utility in healthcare systems, such as EMRs management.

In this work, the architecture of a blockchain based framework applied to EMRs is proposed. The proposed framework aims at providing interoperable, secure, and efficient access to EMRs by health providers, patients and third parties while maintaining the patients' privacy. We propose a timed-based smart contracts whose design meet the demands of EMRs. These contracts are employed in the blockchain for governing the transactions, monitoring the computations performed on the EMRs through the enforcement of the acceptable usage policies and managing the use of data after transmission. Advanced cryptographic techniques are also adopted by the proposed framework for providing further security. In addition, since a medical record is a patient's asset and not a cryptocurrency or a digital currency to be exchanged, unlike previously proposed blockchain based systems for EMRs, we propose a new incentive mechanism that leverages the degree of providers nodes from the perspective of EMRs systems by measuring their efforts regarding maintaining medical records and creating new blocks. Providers' nodes with less degrees are more likely to be selected for creating the new block. As most of the current healthcare systems are welfare oriented that have no intend to involve any monetary value, our proposal rewards the "block's creator" an incentive that will added to its degree to decrease its probability of re-creating the next block instead of just creating a digital currency. Thus, achieving a fairness among providers and ensuring the sustainability of the system.

Moreover, we measure the performance of our proposed system (i.e. average response time, throughput and communication overhead) by conducting analyses on the EMRs' queries. Results show the efficiency of our proposal in handling a large dataset at low latency. In summary, this research presents the design of a blockchain based system for EMRs that handles the issues of privacy, security, data fragmentation, data isolation, effective access to medical records, and system interoperability. The primary contributions of this work are fourfold:

- We provide a complete analysis regarding how the proposed MedChain system and the timed-based smart contracts can interact with the various demands of health providers, patients and third parties.
- We demonstrate how the proposal would address the long-standing issues of privacy and security in the healthcare industry.
- We propose an incentive mechanism that aims at evaluating the degree of health providers regarding their work in maintaining EMRs which in turn will enhance data quality for EMRs.
- We conduct extensive experiments to evaluate the performance of proposed frameworks on various aspects, including throughput, response time, and communication overhead.

II. BACKGROUND

A. BLOCKCHAIN

In 2008, Nakamoto [18] introduced in a pseudonymous paper the Bitcoin cryptocurrency in which the blockchain technology was the key core technology behinds it. The blockchain was first introduced through Bitcoin and designed for maintaining a financial ledger; however, extending its applications to non-financial use cases, such as managing EMRs is a mission for researchers [16]. The blockchain is a distributed database solution that stores a continually increasing set of data records verified and confirmed by participants. In order to achieve the status of consistent consensus system without the need for a trusted third party, several existing distributed computing techniques [19], cryptography, and game theory are adopted by the blockchain technology. Essentially, a block is a data structure including: a block header that includes a hash value of the previous block, timestamp as well as a Merkle root, and a data part that has relevant transactions' data. All of the blocks are linked by the order of the hash value. In the blockchain, the blocks' chain is duplicated across the distributed blockchain network and stored by minors' nodes.

B. SMART CONTRACT

In 1997, Szabo [20] introduced the concept of a smart contract as a mean to digitally formalize and secure relationships over a network. A Smart contract is defined as an application that runs on the blockchain network and is executed by all network participants [21]. Smart contracts are computer codes that govern the blockchain transactions and define the conditions of mutually agreed contracts [22]. Recently, many blockchain based projects have implemented smart contracts, such as the Ethereum platform and Hyperledger. They allow trusted agreements and transactions to be rendered among distinct, anonymous entities with no need for a central authority or external enforcement mechanism. The Ethereum platform allow the developing of smart contracts that suit the requirements of the desired system. In the context of adopting smart contracts in EMRs systems, they allow the creation of scalable and dynamic conditions, terms and rules to securely exchange and sharing medical records.

III. RELATED WORK

Although the blockchain technology was first introduced through Bitcoin, extending its applications to non-financial use cases is an attractive mission for researchers. The industry of healthcare is one of the fields where the blockchain technology is expected to have significant impact. The adoption of the blockchain in the healthcare field can be found in [4]–[13], [15] and [23].

MedRec [7] is a decentralized EMRs management system using the blockchain technology. MedRec is a modular design that manages permissions, authorization and data sharing between participants. The authors highlight the ability of the MedRec to encrypt outside data and preserve hash pointers to patients' health records along with their access permissions in the blockchain. However, access control policy that allows third-parties, such as researchers to access medical data are not explicitly explained. Moreover, MedRec framework incentivizes health providers and medical researchers to participate in mining by earning an Ether, which is an Ethereuem based currency unit for funding continuation of their activities. In other words, they participate in mining to get beneficiaries from the network although the majority of the current healthcare systems are welfare oriented with no intend to involve any monetary value. Similar to MedRec, Ancile [12], which is an Ethereum-based blockchain, is another record management system that utilizes smart contracts for heightened access control and obfuscation of data. Ancile keeps the patients' medical records in the existing databases of providers, and reference addresses to these records along with its permissions are stored in the blockchain network. Ancile allows the interaction between participants and the blockchain. It proposes precise authority levels to each participant node. They propose the access control policy to allow third-party researchers to access medical data. However, the authors do not provide details about their consensus protocol and incentive mechanism. Additionally, no experimental results are explained in the paper. The authors only perform comparative performance analysis by comparing the estimated computational costs of Ancile and MedRec.

Xia *et al.* introduced the BBDS [24] framework that allows owners and participants to access EMR from a shared repository upon successful verification of their identities and keys. BBDS employs the identity-based authentication and key agreement protocol proposed in [25] to provide user membership authentication. However, their secure sharing of sensitive medical information is limited to invited and verified users. Moreover, their proposal of using asymmetric encryption algorithms to encrypt medical information does not seem to be a good option considering the encryption/decryption performance of asymmetric encryption. Fan *et al.* propose the MedBlock [26] which is a hybrid blockchain-based architecture to secure EMRs. According to their architecture nodes are classified as endorsers, orderers and committers. Its consensus protocol is a variant of the PBFT consensus protocol. However, the authors did not explicitly explain the access control policy to allow third-party researchers to access medical data. In [27], Genestier et al. introduce a new idea of utilizing the blockchain to reshape the consent management in the healthcare systems to allow users controlling their whole health record data. However, there is no authorization design and no access control in their implementation.

IV. MEDCHAIN ARCHITECTURE

In this section, we explain the architecture of the proposed MedChain framework in details. The abbreviations used in this paper is summarized in Table. 1.

TABLE 1. Abbreviations used in the paper.

Abbreviation	Meaning
EMR	Electronic Medical Record
PoA	Proof of Authority
REM	Records Evaluation Manager
NCC	Nodes Consensus Contract
SRHC	Steward-Relation History Contract
PRC	Participants' Record Contract
LC	Logs Contract
ACC	Access Control Contract
PReC	Proxy Re-encryption Contract
L	Legibility
CM	Completeness
CR	Correctness
CN	Consistency
NR	Non-Redundancy

A. OVERVIEW

In this section, an architecture that will be built above the existing health providers' databases will be detailed. To reduce the requirements of storing the patients' EMRs in the blockchain and to utilize the existing systems, EMRs will be continuously stored in the providers' databases. As health providers currently maintains and manages the EMRs, while patients can only read data, providers' nodes in our design will be responsible for the maintenance of the blockchain. All accesses to the EMRs will be performed through the blockchain, and accordingly the history of those accesses will be stored in the blockchain to provide a full view of all events occurred to EMRs. Thus, ensuring the integrity of data and preventing misuse of a patient EMR. All logs details in addition to the record ownership metadata will be added to the chain.

Our proposed framework employs the hashing methods, i.e. SHA-256, to ensure data integrity. MedChain keeps a hash value of the link that will be created during the record's issue to access the EMR in the blockchain instead of keeping the link itself. To access a record, the encrypted query link will be sent over HTTPS to the associated participant who has access rights. Therefore, its hash value stored in the blockchain ensures that no alterations have been made outside the blockchain during the transfer as the value of the hash is unique to the original document. For further security, MedChain will store the query link, the key and the EMRs in different locations. Privacy is maintained in the MedChain by employing timed-based smart contracts for governing transactions. Security and access control are maintained by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are provided by the use of comprehensive logs. For crating, validation, and appending new block, the proposed system employs a new incentive mechanism integrated with the Proof of Authority (PoA) consensus algorithm.

B. PRELIMINARIES

1) INCENTIVE MECHANISM

In this paper, we propose a new incentive mechanism integrated with the PoA consensus algorithm to leverage the degree of providers from the perspective of EMRs systems by measuring their efforts regarding maintaining EMRs and creating new blocks. A medical record is a patient asset and not a digital currency or a cryptocurrency to be exchanged. Thus, the degree of a node indicates how significance a provider node owns regarding its quantity and quality of medical records. According to our proposal quality in medical records is defined as having the attributes of legibility, completeness, consistency, correctness, and non-redundancy. The total quality of all EMRs for all users stored in the provider database evaluate the degree of a node.

Providers' nodes with less degrees are more likely to be selected for creating the new block. The node with the least degree will be classified as ‘‘a block's creator'' node, while the nodes with degrees greater than the average degrees of the network will be considered as ‘‘voters''. Voters' nodes are responsible for the validation process when adding new nodes to the system. They validate whether the ID is suitable with the requested role and guarantee that the node is a legitimate health provider or third party. Accordingly, decreasing the possibility that illegitimate nodes can join the system.

As most of the current healthcare systems are welfare oriented that have no intend to involve any monetary value, our proposal rewards the ‘‘block's creator'' node an incentive that will be added to its degree for potential reducing its probability of re-creating the next block instead of just creating a digital currency; thus, achieving a fairness among providers and ensuring the sustainability of the system.

Our proposal plays a key role in improving the data quality of EMRs as providers that 1) fill more legal, correct, consist, complete and no redundant items to an existing record, 2) create new legal, complete, consistent, correct with no redundancy records, and 3) generate a new block will have their degrees increased. Accordingly, they will have less probability to perform the computational task of creating the new block.

2) PROOF OF AUTHORITY (PoA)

The blockchain is a decentralized distributed system that is developed to provide security, privacy, immutability and transparency. All transactions in the Blockchain are completely verified and secured despite the absence of central authority. The reason behinds this is the presence of consensus algorithms. A consensus algorithm is defined as the procedure that is responsible for reaching a common agreement among all nodes in the Blockchain network about the current state of the distributed ledger; thus, achieving reliability and trust among unknown nodes in a distributed computing environment. Basically, the consensus algorithm ensures that any new block, which is added to the Blockchain network, is the one and only version of the truth that is agreed upon by all the nodes in the Blockchain.

Numerus types of proofs, such as the Proof of Work (PoW), and the Proof of Stake (PoS) are employed in different blockchain based systems to determine the miner's block to be appended next. The PoA is a consensus algorithm proposed by Gavin Wood, Ethereum co-founder and former CTO, in 2017 as a replacement for the PoW. It can be used for setting a private blockchain by considering the value of participants' identities to create a set of ‘‘authorities'' that are allowed to create new blocks and secure the blockchain network. In other words, block validators ‘‘authorities'' that are arbitrarily chosen as trustworthy entities are not staking coins but their own reputation instead to maintain security. According to the PoA, verifying the blocks and transactions by authorities who act as moderators of the system achieves several benefits: maintaining the privacy of the system while acquiring the benefits of the blockchain technology; improving the security of the system; minimizing the intensive of computations, increasing the system performance as it provides lower transaction acceptance latency and steady time intervals for issuing blocks.

3) PROXY RE-ENCRYPTION SCHEMA

This schema employs a proxy to solve the problem of transferring encrypted messages among nodes with no need to share symmetric key. It is the responsibility of the proxy to re-encrypts a message in a fashion that allows another user to decrypt it via his/her private key despite the fact that the associated public key is not used for encrypting the message.

Similar to [12] and [28], our proposed system adopts the proxy re-encryption schema to transfer a record from one health provider who stored the record to another without sharing the credentials information used to decrypt the record.

In this work, MedChain adopts the distributed ElGamal re-encryption schema with distributed blinding [29] where a master public key is used to encrypt a message and the associated private key is distributed in pieces to the set of proxies. Thus, the set of proxies will not be able to decrypt the whole message instead they only can re-encrypt that message. According to ElGamal re-encryption schema [29], a set of proxy nodes should be configured in the system such as each node has a unique public/private key pair with the public key known to the other proxy nodes. Also, all proxy nodes have a master public key, and the associated private key is distributed in pieces among the proxy nodes. As shown in Fig. 1, each proxy node will blind the encrypted message by a random blinding factor using ElGamal homomorphic multiplication, then decrypt it using its private key. The result of the decryption will be un-blinded using the original random blinding factor to create obscured plaintext unless determining the blind value. Message decryption could be only done via the intended receiver.

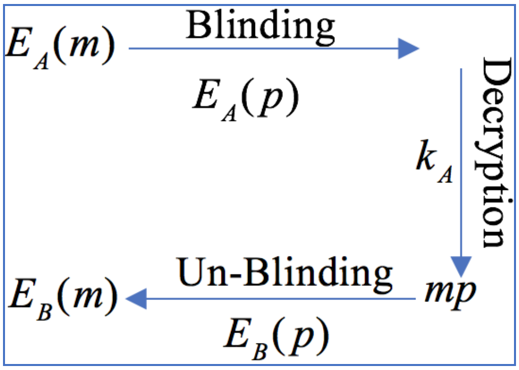

FIGURE 1. Re-encryption using blinding [29].

C. SOFTWARE COMPONENTS

This section details the software components of the MedChain system (see Fig. 2).

1) RECORDS EVALUATION MANAGER (REM)

REM is a python-based tool that evaluates the degree of providers' nodes in the blockchain during the initialization stage. It will be installed and configured only on the providers' nodes. The evaluation process is based on the quantity and the quality of medical records stored in each provider database. To prepare a medical record for quality evaluation, REM, first, extracts features, manages relevant data, and classifies the un-structured parts of that record. In this work, a classification schema that integrates lexical, semantic and syntactical analysis of the record will be employed [30]. Then, the degree of node can be computed as given in equation 6. The degrees of health providers' nodes will be stored in the Nodes Consensus Contract (NCC) to be used for determining voters' nodes and selecting the node to generate the next block.

2) DB MANAGER

Our proposed system integrates the existing medical records stored in the providers' databases by creating links to that records. The DB manager, which is an API written in GoLang, provides an access to the providers' existing databases and is controlled by the permissions information stored in the blockchain. The DB manager functions to navigate the existing database, and create a query link for a patient's medical record. To ensure data integrity, the DB manager creates a hash value for the created query link as well as the patient's medical record to be stored in the Participants' Record Contract (PRC) in the blockchain. It also creates a hash value for the log to be stored in the Logs Contract (LC).

3) CIPHER/DECIPHER MANAGER

This component functions the encryptions and decryptions schemas in our proposed system. Three encryption schemas are utilized in our proposal. Medical records are encrypted using the symmetric key encryption schema. The Cipher/Decipher Manager first generates a symmetric key to encrypt the medical record, and then re-encrypt that key with the public keys of the: provider node, patient node, and the set of proxy nodes. Encrypting the record via the symmetric key encryption schema improves the efficiency and reduce the need for later re-encryption.

To securely distribute information among parties over HTTPS, the public key encryption schema is employed. Moreover, the proxy re-encryption schema is utilized to facilitate an access by a third party.

4) ETHEREUM CLIENT

The Ethereum client is the access point to the Ethereum network as it includes all the functionalities required to join that network [31].

Our design works on a permissioned blockchain network; therefore, nodes with permissions will use the client to access the private blockchain. For the implementation of our proposed prototype, the GoEthereum client is used. It can be accessed by the use of JSON RPC endpoints on the Internet [32]. With GoEthereum, users can access their nodes' information over HTTPS using a wallet that may have different functionality based on the type of node.

5) EMRs INTERFACE

EMRs Interface is the web-based interface that is used for managing EMRs by providers, viewing the EMRs by patients and managing the retrieval options as well as the data sharing. It employs a web3.js library functioning through RPC calls. It collaborates with the DB manager to deliver data form the existing databases and provide users with the update notifications.

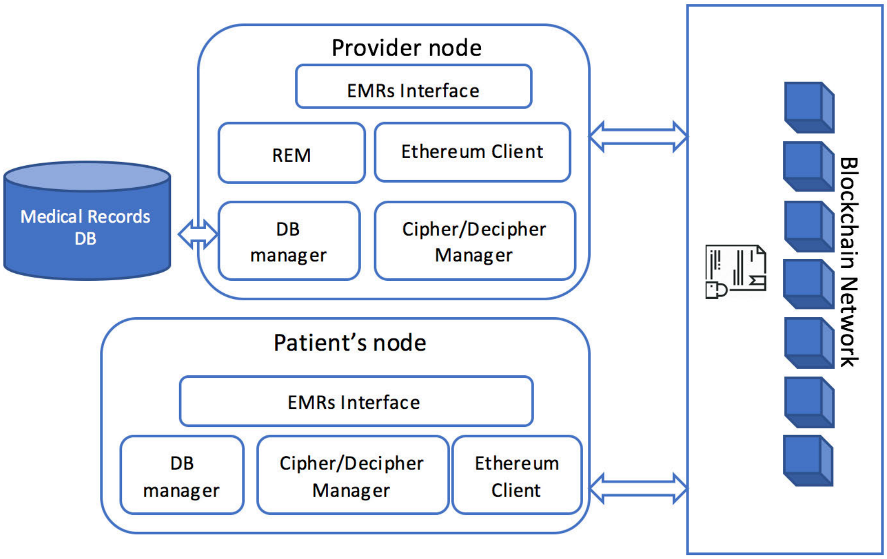

FIGURE 2. Software components of the MedChain system.

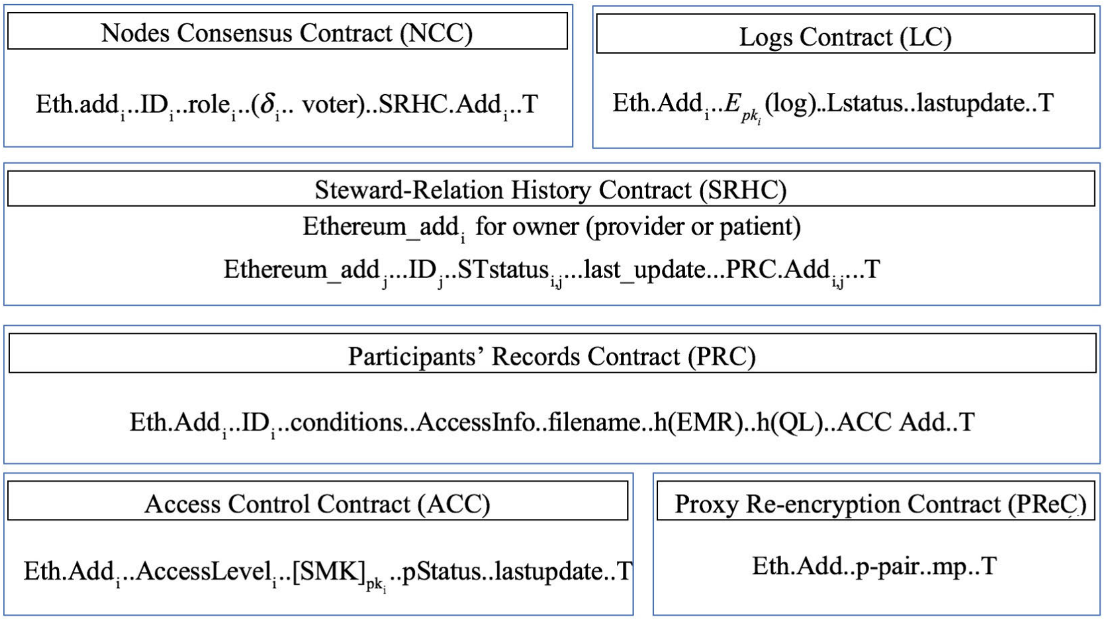
Nodes Consensus Contract (NCC)
$\text{Eth.add}_i..\text{ID}_i..\text{role}_i..(\delta_i..\ \text{voter})..\text{SRHC.Add}_i..\text{T}$
Logs Contract (LC)
$\text{Eth.Add}_i..E_{pk_i}(\text{log})..\text{Lstatus..lastupdate..T}$
Steward-Relation History Contract (SRHC)
Ethereum_add_i for owner (provider or patient)
$\text{Ethereum\_add}_j...\text{ID}_j..\text{STstatus}_{i,j}...\text{last\_update...PRC.Add}_{i,j}...\text{T}$
Participants' Records Contract (PRC)
$\text{Eth.Add}_i..\text{ID}_i..\text{conditions..AccessInfo..filename..h(EMR)..h(QL)..ACC Add..T}$
Access Control Contract (ACC)
$\text{Eth.Add}_i..\text{AccessLevel}_i..[\text{SMK}]_{pk_i}..\text{pStatus..lastupdate..T}$
Proxy Re-encryption Contract (PReC)
Eth.Add..p-pair..mp..T

FIGURE 3. The proposed smart contracts.

V. SMART CONTRACTS

To govern and monitor transactions, a blockchain based system should have its own smart contracts. Our proposed smart contracts employ a set of connectivity and timing functions to provide reasonable period of time for performing transactions and thus ensuring an authorized transaction is intended. All contracts have "T" date field that is used for implementing the timing functions. In our proposed system, as shown in Fig. 3, our smart contracts have:

A. NODES CONSENSUS CONTRACT (NCC)

This contract is considered as an overall contract for preserving the mining, the registration, and certain overwrite procedures for the blockchain. This contract determines voters' nodes in the system along with the provider "block's creator" node who will generate the next block.

It maps a node identification string to its associated Ethereum address identity, i.e. public key. The using of the identification strings rather than the public key directly is to allow already existing IDs to be used. Procedures and Policies coded into the NCC will regulate adding and registering new IDs. This contract additionally determines the role of each participant node within the system i.e. health provider, patient, third party, etc. in order to recognize nodes that have already registered and thus avoiding the case of double registration. For those nodes with a "health provider" role, additional data fields will be appended to maintain the degree of the node along with a Boolean data field that determines whether a provider node is considered as a voter node or not. Additionally, the NCC maps the Ethereum address of the node with its associated SRHC address.

- For determining the voters' nodes, the NCC stores the degree of each provider node within the system that in turn will be used for calculating the average degree of the network and selecting voters' nodes.
- For selecting the ''block's creator'', the NCC assigns this task to the provider node who has the least degree among all providers' nodes in the system.
- For mining, which is defined as the process of adding transactions to the blockchain network, the NCC functions using the PoA Consensus algorithm [33] integrated with our proposed incentive mechanism.
- For the registration process, voters' nodes in the NCC are responsible for validating other nodes who demand a role with ''higher levels'' upon added to the system.
- Voters' nodes guarantee that no threat will be created to the system. Generally, during the blockchain initialization stage, the NCC will be empty. An administrative node (i.e. temporary node) will be added as an initial health provider node and will be removed once there are enough nodes joined the system.
- For the overwriting procedures (i.e. for nodes who may leave the system, such as a provider node goes out of business or those nodes that may harm the system), the NCC can be used to perform the overwriting procedures to revoke permissions associated with that nodes. The overwriting procedure is accomplished by submitting a request to remove a node from the system via a voter node and reaching the majority of votes by the rest of voters. This would then require overwriting the type of a node as terminated, removing it from the NCC, and deleting its related information from the various contracts.

B. STEWARD-RELATION HISTORY CONTRACT (SRHC)

This contract maintains the steward relationship history of each participants' nodes in the system where the patient's medical record is stored and managed by the provider node. The SRHC locates the history of participants medical records by holding a summary list of the steward relationship. For example, if a node role is patient, its SRHC will have references to all health provides that it has been engaged with. On the other hand, the SRHC of a health provider has references to all patients' nodes in which that provider serves. Every node within the blockchain system will have a SRHC that will be created during the registration process.

Generally, the SRHC is identified by the Ethereum address of the SRHC owner node and stores the Ethereum addresses of all associated nodes, their related IDs, a stewardship statue, a last-update date field that indicates the last update on the status field, and an address to the applicable PRC. Users notifications can be enabled via the use of the stewardship status field, such as the stewardship is ''newly'' established, ''awaiting pending updates'', and ''acknowledged patient approval'' or ''acknowledged patient denial''. The stewardship status in the patient SRHC is set by providers node in our system every time they update the patient record or as a part of establishing a new stewardship. Thus, patients can be notified upon modifying the stewardship status field as a new stewardship is recommended or an update is available.

C. PARTICIPANTS' RECORDS CONTRACT (PRC)

This contract aims at tracking all records which health providers store for patients and is generated when a new steward relation is established between two nodes. The PRC includes several data fields with different purposes, and is identified by the Ethereum address of the owner that signifies the patient who owns the listed record. Each record has a filename f, conditions, and AccessInfo. The filename indicated the identity string for the patient record. The condition data field is illustrated for each listed record to indicate special conditions associated with the record, such as a parent can be the owner of the record related to his/her children until they come of age. The condition field also can be appended with a date field to indicate that the record's ownership has to be modified at that time. The AccessInfo data field of the record specifies the needed information to find the EMR Database of a provider, i.e. the provider's host name and the information for the port in a standard network topology. Moreover, to maintain data integrity, each record has a hash value h(QL) for the query link of the file, and a hash value h(EMR) of the stored record. Moreover, a reference to the AAC address is listed in the PRC.

D. LOGS CONTRACT (LC)

This contract tracks all transactions performed on the patients' records to facilitate adding/validating/appending blocks in the blockchain network. This contract is identified by the Ethereum address of the source of the transaction. It lists the transaction details in an encrypted log data field with a status field that indicates whether the new log has been added to the blockchain. It also stores the last update of the status field.

E. ACCESS CONTROL CONTRACT (ACC)

The ACC includes all permissions related information which is specific to every record. It lists the Ethereum addresses for all nodes who have access permissions on the record. This contract specifies the level of that access (i.e. owner, read, and blind-read), and a symmetric key encrypted with the public key of each node. A ''read'' access level indicates that a node (whether a patient node, other provider, or third party) can read the EMR as it has the symmetric key that is generated to encrypt the record when it is first added or the node gets the symmetric key through proxy re-encryption. The blind-read level indicates the PRC can retrieve the symmetric key encrypted for the proxy nodes. The owner level is assigned to the provider node who adds the record. It indicates that a node has full access and control of the ACC as it can add other nodes with the "read" level, remove nodes from the AAC, and also alter the level for any existing nodes. The AAC also contains the "pstatus" field along with the lastupdate in order to notify participants when there is a change in their access level.

F. THE PROXY RE-ENCRYPTION CONTRACT (PReC)

This contract functions the proxy re-encryption schema proposed by [29]. According to Zhou schema, a master public key along with a shared private key will be given to the set of proxy nodes. The PReC will be automatically created whenever establishing a new set of proxy nodes. Each proxy node has a unique public/private key pair with the public key known to others. Each proxy node will select a random blinding value p, encrypts it, blinds the encrypted message with the with encrypted blinding value, and then decrypts portions of blinded message on their own systems. Each proxy node will send its contribution to the PReC that stores the Ethereum address of the proxy node, the encrypted pairs of p values, and the blinded plaintext message "mp".

VI. IMPLEMENTATION OF THE PROPOSED BLOCKCHAIN SYSTEM

A. BLOCKCHAIN INITIALIZATION – PART I: ADDING A NEW HEALTH PROVIDER NODE TO THE BLOCKCHAIN NETWORK

In this stage, all health providers who accept to join the blockchain network have to share their EMRs. All health providers who would like to join the blockchain system should agree on: the rules of the proposed smart contracts, the proposed incentive mechanism, the frequency of updating the blockchain network and the process of generating, verifying, and appending a new block to the blockchain network. In practice, each health provider has an identification string or a public identifier (ID) that must be unique to the health organization. Also, it should be assumed that the Ethereum address (i.e. the Ethereum address is equivalent to a public key) of all providers' nodes that agree to join the blockchain network have been received, and the software components have been installed.

The process of adding a new health provider node starts when the ID and the Ethereum address of the new provider node is sent to the NCC in addition to the requested type role. Voters' nodes in the NCC validate and authenticate that received request by ensuring and confirming that the received request is related to a legitimate health provider who is not registered previously. If the request is accepted and validated, the NCC updates its local memory with the Ethereum address of the new node, its ID and role. The NCC creates a new SRHC for the new provider node whose address will be sent to that provider node.

B. BLOCKCHAIN INITIALIZATION – PART II: COMPUTING THE DEGREE OF A PROVIDER NODE

The process of computing the degree of a node is performed by a node who has a "provider" role listed in the NCC.

In this stage, the REM installed in each health provider node computes the degree of the associated provider node within the network. The degree of a node is calculated based on the quantity and the quality of the EMRs stored in its database. Based on the purposes and the perspective of the designed system, various ways and several attributes could be used to define the quality of medical records. Generally, the quality of a medical record should be judged by whether or not that record serves the purpose for which it was intended. According to [34], a medical record should include: the patient's personal identification information along with his/her medical history. It also should include the medical history of the patient's family, and the patient's medication history, such as name, dosage form, dose, dispensed quantity, and dispensing date. Moreover, the patient's treatments history as well as his/her medical directives have to be included.

To define a measurable standard for the quality in medical records, our proposal considers five key attributes that should be evaluated for each item included in the record. Thus, quality in medical records is defined as having the attributes of legibility, completeness, consistency, correctness, and non-redundancy.

- Legibility (L): Any entry in a medical record have to be legible, dated, timed and authenticated by the health provider [35].
- Completeness (CM): A medical record is considered complete if it has all the above-mentioned items 34], [36], and [37].
- Correctness (CR): The medical record's correctness refers to the accuracy of its collected data. It means that the data provided by health providers should be reflected by the medical record [38], and [39].
- Consistency (CN): A medical record is considered consistent when the included data are reliable, and the data integrity has not been corrupted regardless of how often or in what way the data have been retrieved, viewed, stored, or processed [39].
- Non-redundancy (NR): Redundancy in a medical record indicates that the data of a medical record may be repeated by several health providers.

Thus, by taking the previous attributes into consideration, we define the degree of a health provider δ_i as the total quality of all EMRs for all users stored in the database of that provider.

$$\delta_i = \sum_{EMR=1}^{m} Q_{EMR} \tag{1}$$

where the quality of an EMR is defined as the product of its L, CM, CR, CN and NR attributes.

To compute the Legibility indicator, each item in the EMR will be checked whether it is considered as a legal item or not.

The classification process will be performed via the REM component installed on the provider node. Legal items will be tagged with *i1*, while illegal items will be tagged with *i2*. Thus, the legibility of an EMR equal to 1, if all items of the EMR is considered as legal item; otherwise,

$$L_{EMR} = \frac{\sum i1}{\sum i1 + i2} \tag{2}$$

For the NR indicator, the *NR = 1* for an EMR, if all data stored in a medical record is unique and not repeated by any other health provider otherwise the non-redundancy indicator will be divided among the health provides who share that item.

For the correctness, completeness, and consistency indicators, each item in the EMR will be classified via the REM as: *n1*: correct element, *n2*: incorrect element, *n3*: missing element, *n4*: extra element, and *n5*: conflict and reduction element. Thus,

$$CM_{EMR} = \frac{\sum n1 + n2 + n5}{\sum n1 + n2 + n3 + n5} \tag{3}$$

$$CR_{EMR} = \frac{\sum n1}{\sum n1 + n2 + n4 + n5} \tag{4}$$

$$CN_{EMR} = 1 - \frac{\sum n5}{\sum n1 + n2 + n4 + n5} \tag{5}$$

Accordingly, the degree of a health provider *i* is computed by:

$$\begin{aligned}\delta_i &= \sum_{EMR=1}^{m} Q_{EMR} \\ &= \sum_{EMR=1}^{m} L_{EMR}.CM_{EMR}.CR_{EMR}.CN_{EMR}.NR_{EMR} \qquad (6)\end{aligned}$$

For illustration, assume that a health provider would like to join the blockchain network. Assume that all health providers that would like to join the blockchain network have to agree on 35 important items that should be included in an EMR and thus be measured for all EMRs stored in their databases. Suppose that the provider has two EMRs as detailed in Table. 2.

Table. 3 shows how to compute the quality of the two medical records whose details are presented in Table. 2. In Table. 3, the EMR attributes, the quality of a record, and the degree of a provider node are computed.

At the end of the initialization stage, each health provider will have its degree that will be dynamically stored in the NCC. Thus, the NCC will automatically update the average degrees of nodes within the network as well as voters' nodes. Note, a node who has a degree that is greater than the average degree of the blockchain will be considered as a voter node, while the node that has the least degree among the nodes in the network will be assigned the task of generating the next new block.

TABLE 2. Description of two medical records in a provider node.

Indicator	EMR1	EMR2
L	35 items are legal	34 items are legal and one item is not authorized
n1	15 correct elements	13 correct elements
n2	7 incorrect elements	6 incorrect elements
n3	5 missing elements	7 missing elements
n4	3 extra elements	3 extra elements
n5	5 conflict elements	6 conflict elements
NR	30 items are unique 2 items are shared with other 3 HP 3 items are shared with other 2 HP	28 items are unique 4 items are shared with other 3 HP 3 items are shared with other 2 HP

TABLE 3. An example of computing the quality of two medical records.

	EMR1	EMR2	
L	$\frac{35}{35}=1$	$\frac{34}{35}=0.97$	
CM	$\frac{15+7+5}{15+7+5+5}=0.84$	$\frac{13+6+6}{13+6+7+6}=0.78$	
CR	$\frac{15}{15+7+3+5}=0.50$	$\frac{13}{13+6+3+6}=0.46$	
CN	$1-\frac{5}{15+7+3+5}=0.83$	$1-\frac{6}{13+6+3+6}=0.79$	
NR	$\frac{30+\frac{2}{3}+\frac{3}{2}}{35}=0.92$	$\frac{28+\frac{4}{3}+\frac{3}{2}}{35}=0.88$	
Q_{EMR}	0.32	0.24	$\delta=0.56$

C. ADDING A NEW PATIENT NODE

The procedures of adding a new patient node begins when a request is sent by a provider node. The provider node sends the Ethereum address of the new patient node, its ID and the requested role to the NCC for validation. Similar to the process of validating a request sent for adding a provider node, voters' nodes validate and authenticate that received request by ensuring and confirming that the received request is related to a legitimate patient and the non-existence of a registered patient matching that received ID. If the request is accepted and validated, the NCC updates its local memory with the patient's ID, its Ethereum address and a "patient" role. The NCC creates a new SRHC for the new patient node whose address will be forwarded to the provider. The new patient's account information will be sent to the patient node from the provider who forms the request. This step is equivalent to the process of creating new online accounts.

D. REGISTERING A PATIENT NODE

The process of registering a patient can be viewed as an example of generating a stewardship among two different nodes, one stores and manages the data for the other (i.e. a health provider node stores and manages the data for a patient node). A new patient registration process is performed whenever a new patient visits a health provider.

The process begins by ensuring and confirming that the patient node already is a registered node in the blockchain system. The DB manager of the provider node, who provides an access interface to the existing database, sends the patient's Ethereum address along with its ''patient'' role to the NCC for verification. The NCC ensures that the registration process will be accomplished for a patient node who already registered in the blockchain. The NCC returns a Boolean value for confirmation. Otherwise, the process of adding a new patient node has to be completed first. Upon confirmation, the provider node sends the patient information (i.e. the patient's Ethereum address, and its ID) in a transaction to its SRHC. The SRHC confirms whether the patient is a new patient or not.

Upon confirmation, the SRHC of the provider node requests to generate a new stewardship with the patient who can accept or to reject that request. The SRHC of the provider node will generate a new entry with the Ethereum address of the patient node, its ID, ''waiting approval'' status, and last update of the status. Similarly, the SRHC of the patient node will create a new entry with the Ethereum address of the provider node, its ID, ''waiting approval'' status, and last update of the status. When the patient accepts the request, the status field of both the provider and the patient SRHC will be updated with ''acknowledged patient approval'' along with the last_update field. Otherwise, the process will be canceled, the status field will be updated with ''acknowledged patient approval'', and a notification will be sent to the provider node.

After accepting the request and updating the SRHC of the provider, the provider's SRHC creates a new PRC for the new stewardship. The PRC then accordingly fills the patient's Ethereum address, his/her ID in the Owner field and all the provider database related information. Additionally, the owner's special conditions can be added at this step, such as one of the parents may be a minor's data until he/she comes of age. The PRC sends its address to the SRHC of the provider and the patient nodes to update their ''PRC.add'' data field for future reference.

E. ADDING A NEW RECORD VIA A PROVIDER NODE

The procedures of adding a new record starts after establishing a stewardship between the provider and the patient nodes and thus having a shared PRC. First, internal encryption in the provider node begins the process of adding a new record. When a new record is created by a health provider node, that record will be transferred to the DB Manager. It creates a query link (QL) of a free location in the provider existing Database and hashes both the generated query link *h(QL)* and the record *h(EMR)*.

The DB manager forwards the created query link, and the patient medical record to the Cipher/Decipher Manager for encryption. The Cipher/Decipher Manager generates a symmetric key (SMK), encrypts the new record and link with that key and then encrypts that generated symmetric key with the public keys of the: provider, patient and set of proxies.

The Cipher/Decipher Manager sends the encrypted record to the DB manager to store. Also, all other encrypted data will be sent to the DB manager to create a log indicating the creation of the new record since the history of all access will be stored in the blockchain to provide a full view of all events that happened to each record. The hash of the created log will be calculated and stored in the DB manager for block verification later. Thus, ensuring the integrity of data since if any part of the data is changed, all involved nodes will notice the alteration. Then, the log will be sent to the Cipher/Decipher manager for encryption with the public key of the provider node.

The provider node sends the patient's ID to its SRHC that will return the associated PRC address. The provider node then sends the record information (filename of the record, hash value of the query link h(QL), and hash value of the patient's record h(EMR), the encrypted symmetric keys, and the log) to the PRC. The PRC stores the filename of the record, the hash value of the query link, and the hash value of the patient's record. The PRC then creates a new ACC for the record and forwards the encrypted symmetric keys. The ACC auto-creates the access and permissions information for the record, i.e. patient and provider permissions, and then sends its address to the PRC for its reference. On the other hand, the LC updates its entries with the received encrypted log, the associated Ethereum address of the provider node, the ''new log'' status to indicate that the new log has not been added to the blockchain yet, as well as the timestamp of the last status update.

At the end, the encrypted query link is sent to the patient over HTTPS who will store that link in its cipher/decipher manager and will be used when the patient would like to read his/her record. Additionally, when the new record is created the provider node notifies the NCC to updates the associated degree of the provider node. The NCC informs the REM to add the value of the added record to the node's degree and to return it to perform the update.

F. EDITING A RECORD

The provider node sends the patient's ID to its SRHC to retrieve the associated PRC address. Upon receiving the PRC address, the provider node then sends the filename of the requested record and its Ethereum address to the PRC. The PRC forwards the request to the ACC to check whether that received Ethereum address has a permission (i.e. ''owner'' access level) on the requested record or not. If the provider node has a permission, the AAC forwards the provider's encrypted symmetric key to the PRC. The PRC in turns forwards the received key to the provider node.

The cipher/decipher manager in the provider's node first decrypts the received symmetric key using its private key and then decrypts the query link with that symmetric key. The DB manager of the provider's node follows the related query link and then retrieves the encrypted EMR from the database for editing. Note, when a record is modified, its hash value will also be changed [40]. Thus, the DB manager, after modifying the record, calculates the new hash of the modified record. The DB manager sends the patient's ID to its SRHC to retrieve the associated PRC address. The new hash value will be sent to the PRC for updating.

Moreover, the DB manager creates a log indicating the process of record editing, hash the log and then forwards the log to the cipher/decipher manager for encryption. The encrypted log then will be forwarded to the LC. The LC adds a new entry with the received encrypted log, a ''new log'' status to indicate that the new log has not been added to the blockchain yet, and a timestamp indicating the last status update. Additionally, when the editing the patient's record, the provider node notifies the NCC to updates the associated degree of the provider node. The NCC informs the REM to re-evaluate the value of the record and thus updating the node's degree. The REM performs the calculations and returns the new degree to the NCC for updating.

G. READING A RECORD FROM PATIENT NODE

A patient's node sends the provider's ID to its SRHC to retrieve the associated PRC address. Upon receiving the PRC address, the patient node then sends the filename of the requested record and Ethereum address of the patient to the PRC. The PRC forwards the request to the ACC to check whether that received Ethereum address has a permission (i.e. ''read'' access level) on the requested record or not. If the patient node has a permission, the AAC forwards the patient's encrypted symmetric key to the PRC. The PRC in turns forwards the received key with the database access information to the patient node.

The cipher/decipher manager in the patient's node first decrypts the received symmetric key using its private key and then decrypts the query link with that symmetric key. The DB manager of the patient's node follows the related query link and retrieves the encrypted EMR from the provider's database. Since patients can access their nodes via online wallets, records access can be performed by any device with Internet connection. Thus, improving the interoperability of EMRs. Moreover, the DB manager creates a log indicating the process of reading the record, hashes the log and then forwards the log to the cipher/decipher manager for encryption. The encrypted log then will be forwarded to the LC. The LC adds a new entry with the encrypted received log, a ''new log'' status to indicate that the new log has not been added to the blockchain yet, and a timestamp indicating the last status update.

H. REQUESTING NEW ACCESS LEVEL

The provider node sends the patient ID to its SRHC to return the associated PRC address. The provider node sends the filename of the record to the PRC to return the associated ACC address. The provider node then sends the requested access to the ACC.

The AAC in turns updates it status field to ''request new level'' and its last-update field with the timestamp of the last status update. The ACC then reviews the current access level of that provider node. If the requested level is not in the current level, the ACC requests a change in the access level from the file owner (i.e. patient), and updates its status field to ''waiting approval'' as well as its last-update field with the timestamp of the last update.

If the patient accepts the request, the ACC updates the access level for the applicable file with the ''approved'' status and the last-update. Once the request has been approved, the ACC sends a notification to the provider node indicating the process was successfully completed.

I. A PROVIDER NODE READS A RECORD FROM ANOTHER PROVIDER NODE

The process of reading a record that is stored in a provider node from another provider node utilizes the proxy re-encryption mechanism to increase both the accessibility and the security of EMRs systems.

Suppose that there are two health providers nodes A and B, where provider B would like to read a specific patient's record from provider A. Provider B generates a request to read the record first, signs that generated request by its private key for authorization, and then encrypts the signed request with the public key of provider A. Over HTPPs, the encrypted signed request will be sent to provider A. Upon receiving the request, provider A decrypts the request with its private key and then decrypts it with the public key of provider B to ensure that the provider is the one that it claims to be.

First, node A will send the ID of the patient to its SRHC to return the associated PRC address. Provider node A then sends the filename of the record to the PRC to retrieve the associated ACC and LC address.

The provider node A then sends the Ethereum address and the access level request to the ACC. The ACC then forwards the Ethereum address of provider B and its request for the NCC to verify whether provider B is an authorized provider registered on the system. Upon receiving the verification from the NCC, the ACC generates a new entry with the Ethereum address of node B, the Access Level, and the ''request new level'' status, and the timestamp of the last status update. The ACC requests a change in the access level from the file owner (i.e. patient), and updates both its status field to ''waiting approval'' and last-update. If the patient accepts the request, the ACC updates the access level for the applicable file with the ''approved'' status and the last-update. Once the request

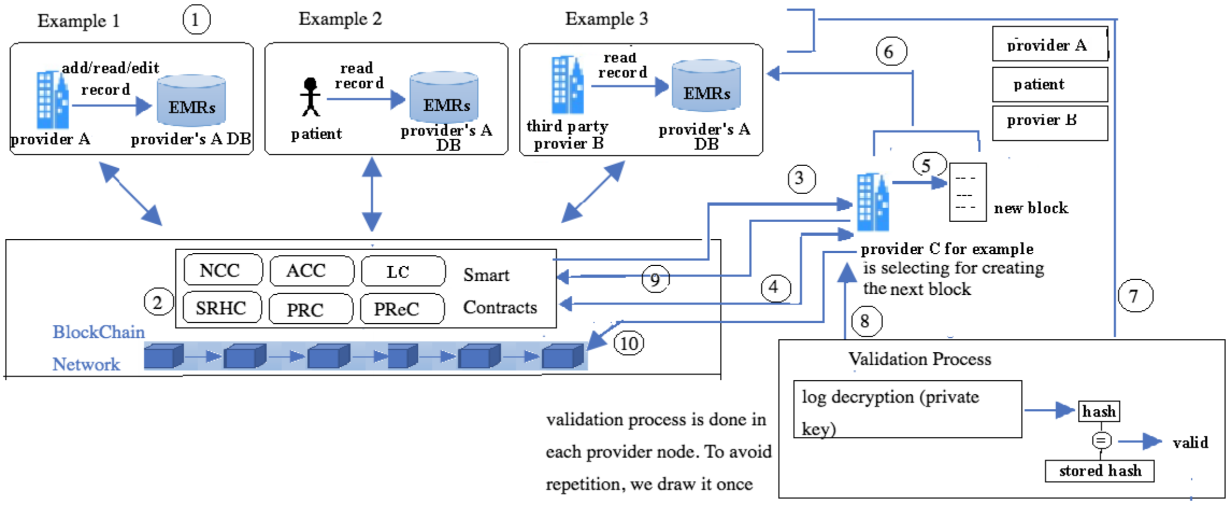

FIGURE 4. Generating/validating/appending a new block.

has been approved, the ACC sends a notification to the provider node B indicating that a new access level is assigned to it. The ACC sends the Ethereum address of provider B and the master encrypted symmetric key to proxy nodes which will re-encrypt that received key. The re-encrypted symmetric key is sent to the ACC that stores it. The ACC sends the ACC address to provider B.

Over HTTPS, provider A sends the encrypted query link to provider B that can decrypt the link and retrieve the record. Provider B has the ACC address that will be used to get the stored ‘‘re-encrypted symmetric key’’. The cipher/decipher manager in provider B decrypts the ‘‘re-encrypted symmetric key’’ using its private key, then decrypts the query link with that symmetric key. The DB manager of provider B follows the query link to retrieve the encrypted record. The record will then be decrypted using the symmetric key.

The DB manager of node B will update the LC entry with an encrypted log indicating the reading of the record. The LC updates it status with ‘‘new log’’ to indicate that the new log has not been added to the blockchain yet, and the timestamp of the last status update.

J. GENERATING/VERIFYING/ AND APPENDING A NEW BLOCK

The process of generating a new block begins by selecting the provider who is responsible for performing this computational task. Based on the degree that each provider node owns, the selection process is performed. According to the selection method in the proposed incentive mechanism, providers with more degrees maintain more medical records or records with higher values. Thus, they are less likely to be selected. The NCC assigns this task to the health provider with the lowest degree among other providers in the system. The process of generating/verifying/ and appending a new block is summarized in Fig. 4.

Step 1: In a daily basis, health providers’ and patient nodes access EMRs. As shown in Fig.4, a provider A access (i.e. read and/or edit) EMRs stored in its database, a provider B on the other hand access EMRs stored in the database of provider A, and a patient reads his EMR which is stored in the database of provider A.

Step 2: All accesses to the EMRs will go through the blockchain and follow the rules of the smart contracts. An encrypted Log will be added to the LC when an access to the EMR is occurred with the status ‘‘new log’’ to indicate that the new log has not been added to the blockchain yet.

Step 3: According to our proposed incentive mechanism, the NCC will assign the task of generating the new block to the provider node who has the lowest degree among the other nodes. As shown in Fig.4, assume that provider C is selected for the process of creating the new block.

Step 4: To generate the new block, provider C sends a request to the LC to return all logs with the status ‘‘new log’’. The LC forwards the request to the NCC to verify that provider C is an authorized provider on the system and it is selected for performing the task of creating the next block. The NCC returns a verification to the LC. Upon receiving the verification, the LC sends all encrypted Logs whose status is ‘‘new log’’ to provider C.

Step 5: After collecting all logs, provider C creates a new block including all logs with the status ‘‘new log’’.

Step 6: Provider C broadcasts to the involved nodes in the blockchain network about the new block and call for verification. Involved nodes are the nodes who consider as the source of the transaction whose log is placed in the LC.

Step 7: Each involved provider node verifies its logs in the new block. Each involved node decrypts the log with its private key, computes the hash value of the log after decryption and compares the computed hash with the value stored in its DB manager.

TABLE 4. Parameters used in the experiments.

Description	Values
The submitted Queries	1,000 – 10,000
The stored EMRs	10,000 – 100,000
Number of nodes	1,000 – 10,000
Transactions in Byte	1,000 – 10,000

Step 8: Each node then sends a signed proof to the provider node C.

Step 9: Upon receiving all the signed proofs, provider C notifies the NCC to update the degree of provider C by adding the incentive value c to its current degree. According to the proposed incentive mechanism, the health provider who will be chosen to generate a new block will get an incentive c as a reward upon successfully verifying the block by other health providers. The value of c depends on the size of the blockchain network and the distribution of health provides degrees. Thus, c will be defined as a fraction of the average degrees in the network.

Step 10: Provider C then broadcasts appending the new block to all providers. After appending the new block, the LC automatically updates the status field for all logs which are added to the chain as ''appended'', and updates the ''lastupdate'' field to indicate the last update on the status field.

VII. EVALUATION AND DISCUSSION

A. PERFORMANCE EVALUATION

1) EXPERIMENTAL SETTING

For evaluation, experiments are performed on a computer system with an Intel Core i7-5557U 3.10 GHz processor, 16 GB memory, and Windows 10 (64 bit) operation system. The Ethereum platform, which is an open source platform featuring smart contract (scripting) functionalities, is used for implementing the proposed system. Our smart contracts are written in Solidity and deployed with Truffles with no capacity restrictions on the stored data size. To allow the interaction with an Ethereum node using HTTPS, the web3.js library is employed. The open source Apache JMeter is used as a functional and performance measurement tool for testing the services provided on the web. A sample anonymized data set, namely OpenMRS, that includes 5,000 patients and 500,000 observations is used for evaluation [41]. The MySQL queries is used first to create the database, import the data, and then deploy the OpenMRS. For designing the evaluation experiment, our previous published research was considered [19] to select the experiment parameters, and the values of that parameters were selected to estimate the approximate number of EMRs based on statistics published by the ministry of health in Palestine where the corresponding author lives [42]. The experiments' parameters, and their values are given in Table. 4.

We performed several tests to compare the performance of our proposed MedChain system with the traditional relational EMRs management system based on two parameters: the number of submitted queries, and the size of the stored medical records. The measurement of the performance was based on the following metrics: the average response time, the throughput, and the communication overhead. Only one parameter was changed each time so that any changes in the performance would be based solely on this parameter. In fact, results achieved from these tests were used to study: (1) the behavior of the proposal for random systems with different number of nodes and roles; (2) the behavior of the proposal for systems with different medical records' size. We also evaluate the round-trip execution time of transactions in the MedChain system and the traditional relational database systems.

To study the effects of changing the distribution of the submitted queries on the average response time, the throughput, and the communication overhead, theses queries were varied from 1000-10,000 queries unit, and the distribution of the submitted queries among the nodes were carried in the following manner.

- 25% variations: Similar requests' distributions among nodes.
- 50% variations: The intermediate situation where the majority of queries are submitted to 50% of nodes.
- 75% variations: The advanced intermediate situation where the majority of queries are submitted to 75% of nodes.

To study the effects of increasing the number of medical records stored in the providers' databases on the average response time, the throughput, and the communication overhead, the number of stored records were varied from 10,000 – 100,000, and the distribution of the records among the nodes were carried in the following manner.

- 25% variations: Similar records' distributions among nodes.
- 50% variations: The intermediate situation where the majority of records are stored in 50% of nodes.
- 75% variations: The advanced intermediate situation where the majority of records are stored in 75% of nodes.

2) RESULTS AND DISCUSSION

Results show that the average response time and the average number of messages sent per node increased as the total submitted queries was increased as shown in Fig. 5 and Fig. 6.

Similarly, the average response time and the average number of messages sent per node increased as the total number of stored records was increased as shown in Fig. 7 and Fig. 8. These situations are expected as the more queries to be submitted, the longer it takes for a query to be completed and the more communications among participants' nodes to be occurred. Also, the more records to be stored, the more participants to use the system; and thus, the more queries to be submitted on the system. However, the throughput of the system remains constant even with the increments of the submitted queries or the increments of the stored records (Fig. 9 and Fig. 10). The stability of the system's throughput

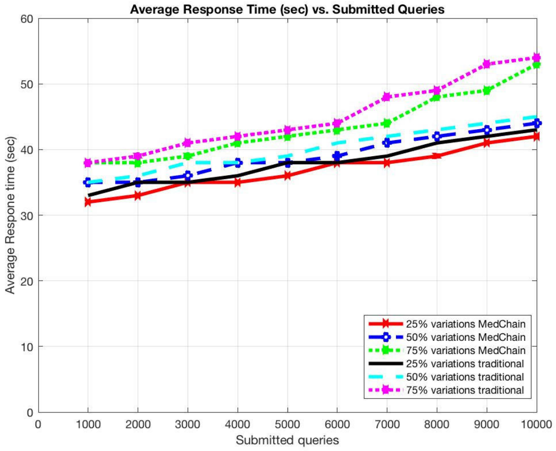

FIGURE 5. Average response time (sec) vs. submitted queries.

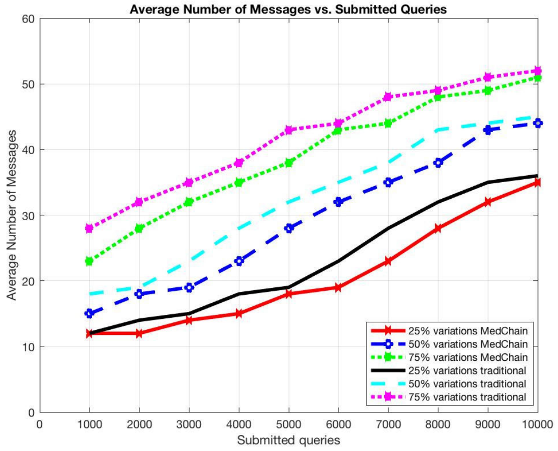

FIGURE 6. Average number of messages vs. submitted queries.

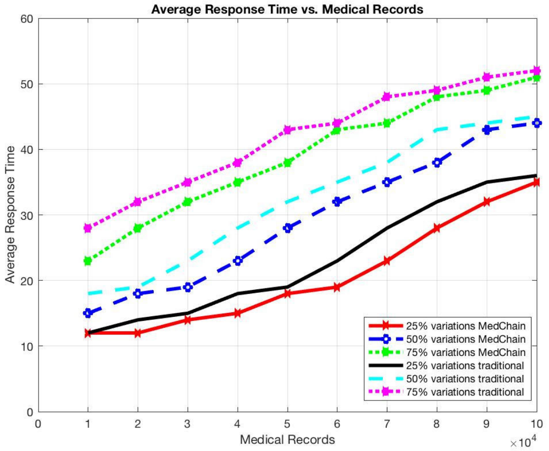

FIGURE 7. Average response time (sec) vs. medical records.

even when increasing the number of stored medical records and the number of submitted queries prove the ability of the system to handle and process a large dataset with high frequency at low latency as in EMRs systems.

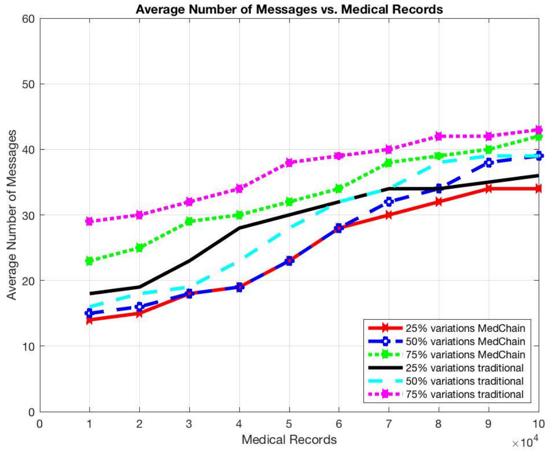

FIGURE 8. Average number of messages vs. medical records.

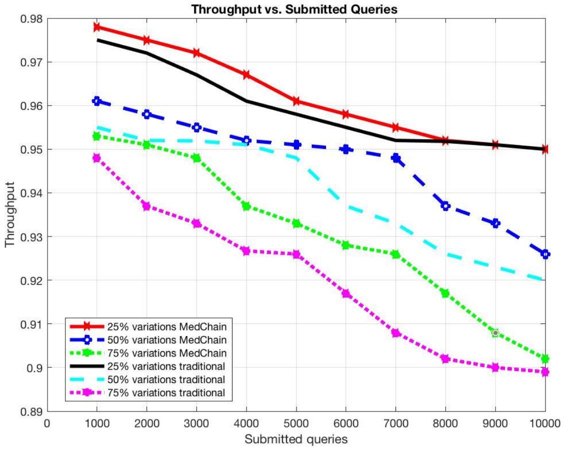

FIGURE 9. Throughput vs. submitted queries.

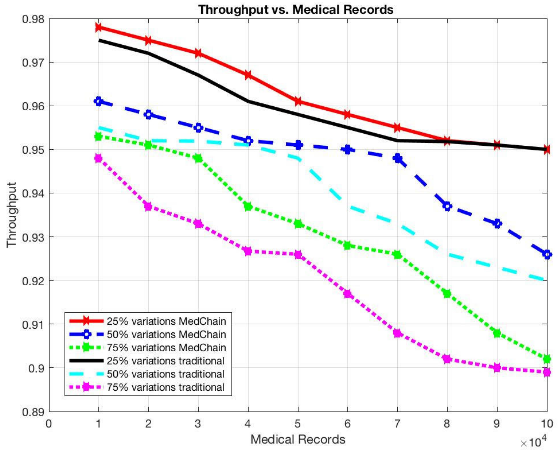

FIGURE 10. Throughput vs. medical records.

We perform a comparison between the traditional relational database management system and the proposed blockchain system regarding the transaction round-trip execution time. The round-trip time is defined as the time in which a transaction's request takes to be sent in addition to the length of time it takes for an acknowledgement to be received by the web client. To perform the experiment, we evaluate the time required to perform query transactions and the time spent on performing invoke transactions. It was obvious to see that the round-trip execution time increases as the number of transactions increases. Also, results show that time spent on performing invoke transactions is more than the time spent on performing the query transactions. The reason behinds this is invoke transactions require endorsement while endorsement is not required for query transactions.

Results show that there is no significant difference between our proposal MedChain system and the traditional EMRs management systems (i.e. relational database systems) in terms of the average response time (Fig.5, Fig.6), the average number of messages (Fig7. Fig.8), the throughput (Fig9, Fig.10) and the round-trip execution time of transactions (Fig.11).

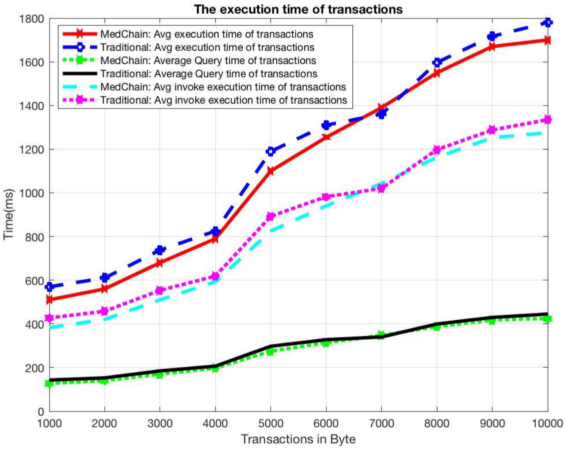

FIGURE 11. **The round-trip execution time of transactions.**

Actions in the proposed system are classified into off-blockchain and on-blockchain actions. The off-blockchain actions involve computing the degree of providers' nodes during the blockchain initialization, creating a query link, calculating its hash value and encrypting it when adding a new record. Also, database storage and retrieval procedures are off-blockchain actions. The on-blockchain actions include retrieving and storing data values in smart contracts, sending internal transactions to link different contracts, and spawning new contracts using other contracts. The diversification of the system modules and the balance between the on-chain and the off-chain actions involve increasing the features of the proposed system while maintaining its performance.

Our system adopts the PoA consensus algorithm which plays a significant role in improving the system performance and decreasing the computational time and cost as it can handle more transactions per second compared with the PoW and the PoS. The computational cost is defined as the cost of validating transactions by each node in the blockchain, while the computational time is defined as the number of steps required for transactions and blocks validations. The PoS involves the use of a stake, which is defined as the number of tokens that each node holds, in deciding who will validate the new block. According to the PoS, nodes with more tokens will be more likely to be chosen as a validator for the next block. Rewards are derived from transaction fees as no new coins are created in the process. The PoW involves the process of mining by performing calculations to verify the legitimacy of transactions and create new currencies. The first miner to complete the calculation is rewarded with a new coin and the block is added to the chain. Compared to the PoW and the PoS, the PoA replaces the security model of the PoW from a financial incentive to trust based. While the PoW mechanism requires mining, PoA is much less intensive in its approach and thus requires less computational power. PoA is significantly faster at processing transactions because of the need for less communication between nodes.

According to our proposal, two rounds are required for the generation and the validation of the new block (i.e. block generating and block acceptance). The set of N trusted nodes who joined the blockchain network are called the authorities. Nodes with degrees greater than the average degree of the network will be considered as ''voters''. Voters' nodes are responsible for the validation process when adding new nodes to the system. The node with the least degree will be classified as ''a block's creator'' node. The degree of a node is employed to fairly distribute the responsibility of block creation among authorities since the node that has the least degree will be responsible for block creation. Block creation and validation needs two rounds. In the first round, the block's creator node collects the transactions placed in the LC with the status ''new log'' to create the new block and sends it to only involved nodes (i.e. no need to send the block to all participants in the network). Then, in the second round (block acceptance), the generated block should be validated by the involved nodes who are considered as the source of the transactions. These nodes validate their logs and then reply with a verification. A block will be appended when receiving all associated verifications. Because the involved nodes only participate in the process of verifying the blocks and transactions, the number of sent messages will be decreased; thus, ensuring low computational cost and time (steps) of the system. Additionally, it maintains the privacy of the system while acquiring the benefits of the blockchain technology, improves the security of the system, minimizes the intensive of computations and increases the system performance as it provides lower transaction acceptance latency.

The adoption of timed-based smart contracts ensures a reasonable period for the transactions and the computations performed on the data. The loss of connectivity resets the timers to zero and the data is destroyed using instructions stored in the smart contracts.

Although previous research concluded that the performance of blockchain limits its applicability in various industries, they recommended that a practical solution may be combining blockchain with relational databases to implement industrial functions [43]. In this work, results indicate that there is no significant difference between the proposed MedChain and the traditional relational system. In our design, the MedChain system is divided into data-intensive and non-data-intensive modules. As mentioned when illustrating the architecture of the system, our system will be built on the top of the health providers' databases. Thus, the advantages of the relational databases regarding its high performance and the advantages of the blockchain regarding the high security and privacy, in addition to the inter-operable and effective access to EMRs will be both utilized. To obtain the best balance between performance and security. Data modules are divided into data intensive and non-data intensive modules. The relational databases will be responsible for the data intensive modules, such as storing the EMRs and retrieving the data. The non-data-intensive modules, such as EMRs permissions are built on blockchain.

Additionally, the trust and security related parts of the MedChain system are simplified in data volume using the hashing techniques to meet the data reading and writing performance.

B. SYSTEM COMPARISON

A comparison between different blockchain-based frameworks for managing EMRs presented in the literature is shown in Table. 5.

As shown in the table, previous studies have proposed blockchain-based systems for sharing and managing EMRs with various supported features that meet the requirements of the desired system. Similar to others work, our main objective is to provide a blockchain-based system for secure, interoperable, and efficient access to EMRs while maintaining the patient's privacy. MedChain meets the HIPAA security requirements and rules. By the proposed smart contracts, MedChain verifies legitimate participants before performing the registration process, employs the identity checking for ensuring that sensitive information is only given to authorized users. MedChain confirms the integrity of data by adopting the hashing techniques and employs the proxy re-encryption to allow a third party to read the EMR stored in one provider node.

TABLE 5. Comparison between proposed frameworks for managing EMRs in the literature.

Properties	[8]	[7]	[12]	[10]	[23]	MedChain
Blockchain-based	√	√	√	√	√	√
Access Control	√	√	√	√	√	√
Privacy Preservation	√	√	√	√	√	√
Scalability	√	√	√	√	√	√
Healthcare Legislation	√	√	√	√	√	√
Smart Contracts	√	√	√	√	√	√
Timed-based smart contracts	X	X	X	√	X	√
Notification based smart contract	X	√	√	X	√	√
Performance Evaluation	X	X	√ (estimating computational costs of [12] and [7]	X	X	√
Incentive mechanism	X	X	X	X	X	√

Unlike previous work, we propose a new incentive mechanism integrated with the PoA consensus algorithm for creating and validation new blocks. According to the several blockchain protocols, an incentive (i.e. digital currency, such as Bitcoin and Ether) will be obtained as a transaction fee or a reward for mining block. The MedRec framework incentivizes health providers to participate in mining to earn an Ether, which is an Ethereuem based currency unit for funding continuation of their activities. Also, MedRec incentivizes medical researchers and health care authorities to participate in mining to get beneficiaries from the network. Since the majority of the current healthcare systems are welfare oriented with no intend to involve any monetary value, there is a need to propose a more suitable incentive mechanism for the EMRs systems. Thus, our proposed framework utilizes the concept of a node degree that evaluates the significance of health providers from the perspective of the EMRs systems by measuring their efforts that have been made on maintaining medical records and creating new blocks. Providers' nodes with more degrees are less likely to be assigned to the computation task of creating the new block. Our proposal aims at achieving the fairness and the equality status among providers' nodes for ensuring the sustainability of the system by rewarding the "block's creator" an incentive that will be added to its degree for decreasing its probability of re-creating the next block, instead of just creating a digital currency.

Our system utilizes the timed smart contracts for controlling accesses to EMRs and monitoring the computations performed on the EMRs through the enforcement of the acceptable usage policies.

We measure the performance of our proposed system, via Apache JMeter, by conducting analyses on EMRs queries, such as creating/reading/updating/sharing records. We present experimental results for average response time, throughput, and communication overhead to prove the efficiency of the proposed system in handling a large dataset at low latency.

C. EMRS ACCESS WITH/WITHOUT BLOCKCHAIN

Generally, an EMR is a record in a database that stores medical information about a patient in a digital format. An EMR system is a digital tool with a web and/or mobile interface that helps users to maintain their health and manage their care by allowing them to capture their own health and care data, to communicate with health and care services, and/or to have access to their medical record. Relational databases are used to implement the existing EMRs management systems.

Currently, a patient may visit more than one healthcare provider, such as general practitioner, specialists, public hospital, private clinics, etc. for various needs. In fact, the record will be stored in the provider's private database who had issued the record and will only be the eligible provider for editing, managing and maintaining that record. In other words, the EMRs of a patient are placed in different health providers' databases, thus providers cannot have a comprehensive overview of all the records of a single patient. Providers' databases are partly open to patients and other health providers with different specified permissions. Patients' with access rights could query their EMRs from different providers. Providers' with access rights could query EMRs of a common patient from other health provider when there is a need, such as consulting related EMRs for making diagnosis. These situations cause a lack of coordinated data management and exchange. Hence, medical records are isolated and fragmented across public hospitals, private clinics, private practices, labs and private companies collecting data from wearable devices rather than cohesive.

Moreover, as health providers are solely maintaining the records in which they had issued, there is a difficulty to confirm data integrity when a malicious entity modifies that single copy of the record or even when a record is removed from a provider database. The need for multiple access to the EMRs had raised the interoperability challenges between patients and health providers which pose additional barriers to effective data sharing.

In addition, as technology is constantly evolving, several advanced techniques are developed to violate digital privacy and security. Unfortunately, medical records are considered as major targets for information theft since they include private and sensitive information, e.g. the patients' names, identity numbers, contacts info and addresses. In 2015, a hack happened on Anthem which is an insurance corporation results in stealing the records of 78.8 million U.S patients [44].

On the other hand, our proposed MedChain system will be built above the existing health providers' databases to facilitate the integration with the existing systems. To reduce the requirements of storing the patients' EMRs in the blockchain and to utilize the existing systems, EMRs will be continuously stored in the providers' databases. As health providers currently maintains and manages the EMRs, while patients can only read data, providers' nodes in our design will be responsible for the maintenance of the blockchain.

The MedChain system employs the blockchain technology which is a collection of techniques (cryptography and hash functions) to create a chain of data (i.e. un-breahable ledger) where each new piece of data is linked to the previous one by a cryptographic hash function. Therefore, it significantly increases the difficulty of attack and improves the privacy and security of EMRs. All accesses to the EMRs will be performed through the blockchain, and accordingly the history of those accesses will be stored in the blockchain to provide a full view of all events occurred to EMRs. Thus, ensuring the integrity of data and preventing misuse of a patient EMR. All logs details in addition to the record ownership metadata will be added to the chain.

Sensitive information that are placed on the blockchain are encrypted to decrease the possibility of being accessed by unauthorized entity. MedChain system increases the level of data obfuscation by separating sensitive information by using SRHC, PRC and ACC. The use of proxy re-encryption technique is employed to solve the problem of transferring encrypted messages among nodes with no need to share symmetric key.

Our proposed framework employs the hashing methods, i.e. SHA-256, to ensure data integrity. MedChain keeps a hash value of the link that will be created during the record's issue to access the EMR in the blockchain instead of keeping the link itself. To access a record, the encrypted query link will be sent over HTTPS to the associated participant who has access rights. Therefore, its hash value stored in the blockchain ensures that no alterations have been made outside the blockchain during the transfer as the value of the hash is unique to the original document. For further security, MedChain will store the query link, the key and the EMRs in different locations.

Privacy is maintained in the MedChain by employing timed-based smart contracts for governing transactions. Security and access control are maintained by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are provided by the use of comprehensive logs.

For crating, validation, and appending new block, the proposed system employs a new incentive mechanism integrated with the Proof of Authority (PoA) consensus algorithm. Our proposal leverages the degree of providers nodes from the perspective of EMRs systems by measuring their efforts regarding maintaining medical records and creating new blocks. Providers' nodes with less degrees are more likely to be selected for creating the new block. As most of the current healthcare systems are welfare oriented that have no intend to involve any monetary value, our proposal rewards the "block's creator" an incentive that will added to its degree to decrease its probability of re-creating the next block instead of just creating a digital currency. Thus, achieving a fairness among providers and ensuring the sustainability of the system.

VIII. CONCLUSION

In this paper, a design of a blockchain based system, namely MedChain, for managing EMRs is proposed. MedChain is designed to be compatible with the existing EMRs' databases and to improve the current EMRs management systems as it provides interoperable, secure, and efficient access to EMRs by health providers, patients and third parties, while maintaining the patients' privacy. In MedChain, the blockchain maintenance including creation, verification and appending of new blocks is the responsibility of health providers, while allowing patients to securely control accesses their EMRs. Privacy is maintained in the MedChain by employing timed-based smart contracts for governing transactions and monitoring the computations performed on the EMRs through the enforcement of the acceptable usage policies. The adoption of hashing techniques ensures the integrity of data. Security and access control are maintained by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are provided by the use of comprehensive logs. Our proposal is independent of any specific system, and its variations can potentially accommodate other similar systems with multiple access for electronic records.

As medical records are patients' assets and not a cryptocurrency or digital currency to be exchanged, this work propose a new incentive mechanism integrated with the PoA for mining. It leverages the degree or significance of providers regarding their efforts on maintaining medical records and creating new blocks. Since most of the current health providers are welfare oriented that have no intend to involve any monetary value, our mechanism rewards the "block's creator" an incentive to be added to its degree and accordingly decreasing its probability of re-creating the next block instead of just creating a digital currency. Thus, achieving the fairness and the equality among providers and ensuring the sustainability of the system. Extensive experiments are conducted to evaluate the MedChain performance on different aspects, including response time, throughput, and communication overhead. Results indicate the efficiency of our proposal in handling a large dataset at low latency.

ACKNOWLEDGMENT

The authors would like to thank Palestine Technical University–Kadoorie, Bank of Palestine, and Taawon association for their support.

REFERENCES

[1] A. R. Rajput, Q. Li, M. T. Ahvanooey, and I. Masood, "EACMS: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84304–84317, 2019.

[2] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[3] L. X. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Gener. Comput. Syst.*, vol. 95, pp. 420–429, Jun. 2019.

[4] L. A. Linn and M. B. Koo, "Blockchain for health data and its potential use in health IT and health care related research," in *Proc. ONC/NIST Blockchain Healthcare Res. Workshop.*, Gaithersburg, MD, USA, 2016, pp. 1–10.

[5] D. Ivan, "Moving toward a blockchain-based method for the secure storage of patient records," in *Proc. ONC/NIST Blockchain Healthcare Res. Workshop*, Gaithersburg, MD, USA, 2016, pp. 1–11.

[6] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–5.

[7] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data*, Aug. 2016, pp. 25–30.

[8] Q. I. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.

[9] K. Culver, "Blockchain technologies: A whitepaper discussing how the claims process can be improved," in *Proc. ONC/NIST Use Blockchain Healthcare Res. Workshop*, Gaithersburg, MD, USA, 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/3-47-whitepaperblockchainforclaims_v10.pdf

[10] S. Amofa, E. B. Sifah, K. O.-B. O. Agyekum, S. Abla, Q. Xia, J. C. Gee, and J. Gao, "A blockchain-based architecture framework for secure sharing of personal health data," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Ostrava, Czech Republic, Sep. 2018, pp. 1–6.

[11] G. Yang and C. Li, "A design of blockchain-based architecture for the security of electronic health record (EHR) systems," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nicosia, Cyprus, Dec. 2018, pp. 261–265.

[12] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018.

[13] X. Zhang, S. Poslad, and Z. Ma, "Block-based access control for blockchain-based electronic medical records (EMRs) query in eHealth," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[14] A. A. Alomar, M. Z. A. Bhuiyan, and A. Basu, S. Kiyomoto, and M. S. Rahman, "Privacy-friendly platform for healthcare data in cloud based on blockchain environment," *Future Gener. Comput. Syst.*, vol. 95, pp. 511–521, Jun. 2019.

[15] L. Hang, E. Choi, and D.-H. Kim, "A novel EMR integrity management based on a medical blockchain platform in hospital," *Electronics*, vol. 8, no. 4, p. 467, Apr. 2019.

[16] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, "Blockchain technology in healthcare: A systematic review," *Healthcare*, vol. 7, no. 2, p. 56, Apr. 2019.

[17] Alexander, "Electronic health records implementation with blockchain, BPM, ECM, and platform," Samarin.Biz, Geneva, Switzerland, Tech. Rep., 2016. [Online]. Available: http://improving-bpm-systems.blogspot.com/2016/07/electronic-health-records-ehr.html

[18] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," Working Paper, 2008

[19] E. Y. Daraghmi and S.-M. Yuan, "A small world based overlay network for improving dynamic load-balancing," *J. Syst. Softw.*, vol. 107, pp. 187–203, Sep. 2015.

[20] N. Szabo, "The idea of smart contracts," Nick Szabo's Papers Concise Tuts., Tech. Rep., 1997, vol. 6.

[21] X. Liu, K. Muhammad, J. Lloret, Y.-W. Chen, and S.-M. Yuan, "Elastic and cost-effective data carrier architecture for smart contract in blockchain," *Future Gener. Comput. Syst.*, vol. 100, pp. 590–599, Nov. 2019.

[22] R. Modi, *Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain*. Birmingham, U.K.: Packt, 2018.

[23] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 267–278, Jul. 2018.

[24] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[25] D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," Ripple Labs, San Francisco, CA, USA, White Paper 5, Feb. 2018, p. 8.

[26] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 136, Aug. 2018.

[27] P. Genestier, S. Zouarhi, P. Limeux, D. Excoffier, A. Prola, S. Sandon, and J.-M. Temerson, "Blockchain for consent management in the ehealth environment: A nugget for privacy and security challenges," *J. Int. Soc. Telemed. eHealth*, vol. 5, pp. e24-1–e24-4, Apr. 2017.

[28] A. Manzoor, M. Liyanage, A. Braeke, S. S. Kanhere, and M. Ylianttila, "Blockchain based proxy re-encryption scheme for secure IoT data sharing," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Seoul, South Korea, May 2019, pp. 99–103.

[29] L. Zhou, M. A. Marsh, F. B. Schneider, and A. Redz, "Distributed blinding for distributed elgamal re-encryption," in *Proc. 25th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Columbus, OH, USA, Jun. 2005, p. 824.

[30] A. Alicante, F. Amato, G. Cozzolino, F. Gargiulo, N. Improda, and A. Mazzeo, "A study on textual features for medical records classification," *Stud. Health Technol. Inform.*, vol. 207, pp. 370–379, Jan. 2014.

[31] (2016). *Ethereum Clients*. [Online]. Available: http://www.ethdocs.org/en/latest/ethereum-clients/index.html

[32] (2016). *JSON RPC*. [Online]. Available: https://github.com/ethereum/wiki/wiki/JSON-RPC#json-rpc-endpoint

[33] N. Prusty, *Building Blockchain Projects: Building Decentralized Blockchain Applications With Ethereum and Solidity*. Birmingham, U.K.: Packt, 2017.

[34] J. Hicks. (2019). *The Basic Components of a Complete Medical Record*. [Online]. Available: https://www.verywellhealth.com/importants-parts-of-a-medical-record-2317249.

[35] G. E. Hall, "Legal aspects of medical records," *JAMA*, vol. 191, no. 10, p. 872, Mar. 1965.

[36] F. W. Lai, J. A. Kant, M. H. Dombagolla, A. Hendarto, A. Ugoni, and D. M. Taylor, "Variables associated with completeness of medical record documentation in the emergency department," *Emergency Med. Australasia*, vol. 31, no. 4, pp. 632–638, Aug. 2019.

[37] T. A. Nguyen, W. A. Perkins, T. J. Laffey, and D. Pecora, "Checking an expert systems knowledge base for consistency and completeness," in *Proc. 9th Int. Joint Conf. Artif. Intell.*, Los Angeles, CA, USA, vol. 1, 1985, pp. 375–378.

[38] J. R. Logan, P. N. Gorman, and B. Middleton, "Measuring the quality of medical records: A method for comparing completeness and correctness of clinical encounter data," in *Proc. AMIA Symp.*, 2001, pp. 408–412.

[39] K. Pantazos, S. Lauesen, and S. Lippert, "Preserving medical correctness, readability and consistency in de-identified health records," *Health Inform. J*, vol. 23, no. 4, pp. 291–303, Dec. 2017.

[40] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2006.

[41] B. Mamlin and W. Luyima. (2011). *OpenMRS Wiki—Demo Data*. [Online]. Available: https://wiki.openmrs.org/display/RES/Demo+Data

[42] *Health Annual Repost in Palestine*, PHIC, Nablus, Palestine, 2016.

[43] S. Chen, J. Zhang, R. Shi, J. Yan, and Q. Ke, "A comparative testing on performance of blockchain and relational database: Foundation for applying smart technology into current business systems," in *Proc. Int. Conf. Distrib., Ambient, Pervasive Interactions*, 2018, pp. 21–34.

[44] N. Akpan, "Has health care hacking become an epidemic," PBS Newshour, New York, NY, USA, Tech. Rep., 2016.

EMAN-YASSER DARAGHMI received the B.S. degree in communication and information technology from Al-Quds Open University, in 2008, and the M.S. degree in computer science and the Ph.D. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2011 and 2015, respectively. She is currently an Assistant Professor with the Department of Applied Computing, Palestine Technical University–Kadoorie (PTUK). Her current research interests include distributed computing, cloud computing, distributed systems, the Internet technologies, and blockchain. She serves as a Technical Program Committee Member for several conferences and a Reviewer for highly distinguished journals.

YOUSEF-AWWAD DARAGHMI received the bachelor's degree in electrical engineering from An-Najah National University, Palestine, in 2002, and the master's degree in computer science and information engineering and the Ph.D. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2007 and January 2014, respectively. He is currently an Assistant Professor with the Computer Systems Engineering Department, Palestine Technical University–Kadoorie. His research focuses on blockchain, intelligent transportation systems, and vehicular ad-hoc networks. He serves as a TPC Member for several conferences and a Reviewer for highly distinguished journals. He received the Best Paper Award from ITST, in 2012.

SHYAN-MING YUAN received the B.S.E.E. degree from National Taiwan University, in 1981, the M.S. degree in computer science from the University of Maryland, Baltimore, in 1985, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1989. He joined the Electronics Research and Service Organization, Industrial Technology Research Institute, as a Research Member, in 1989. Since September 1990, he has been an Associate Professor with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. He became a Professor, in 1995. His research interests include cloud computing, the Internet technologies, and distance learning.