IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 8, 2019, accepted October 2, 2019, date of publication October 15, 2019, date of current version October 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947613

Lightweight Blockchain for Healthcare

LEILA ISMAIL[1], (Member, IEEE), HUNED MATERWALA[1], AND SHERALI ZEADALLY[2]
[1]College of Information Technology, UAE University, Al Ain 15551, UAE
[2]College of Communication and Information, University of Kentucky, Lexington, KY 40506-0024, USA

Corresponding author: Leila Ismail (leila@uaeu.ac.ae)

This work was supported by the Emirates Center for Energy and Environment Research of the United Arab Emirates University under Grant 31R101.

ABSTRACT Healthcare data management has been gaining a lot of attention in recent years because of its high potential to provide more accurate and cost-efficient patient care. The traditional client-server and cloud-based healthcare data management systems suffer from the issues of single point of failure, data privacy, centralized data stewardship, and system vulnerability. The replication mechanism, and privacy and security features of blockchain have a promising future in the healthcare domain as they can solve some of the inherent issues of the health management system. However, most of the recent research works on blockchain in the healthcare domain have primarily focused on the permission-less Bitcoin network that suffers from drawbacks such as high energy consumption, limited scalability, and low transaction throughput. Consequently, there is a need for a scalable, fault-tolerant, secure, traceable and private blockchain to suit the requirements of the healthcare domain. We propose a lightweight blockchain architecture for the healthcare data management that reduces the computational and communication overhead compared to the Bitcoin network by dividing the network participants into clusters and maintaining one copy of the ledger per cluster. Our architecture introduces the use of canal, that allows secure and confidential transactions within a group of network participants. Furthermore, we propose a solution to avoid forking which is prevalent in the Bitcoin network. We demonstrate the effectiveness of our proposed architecture in providing security and privacy compared to the Bitcoin network by analyzing different threats and attacks. We also discuss how our proposed architecture addresses the identified threats. Our experimental results demonstrate that our proposed architecture generates 11 times lower network traffic compared to the Bitcoin network as the number of blocks increases. Our ledger update is 1.13 times faster. Our architecture shows a speedup of 67% in ledger update and 10 times lower network traffic when the number of nodes increases.

INDEX TERMS Blockchain, consensus, decentralization, health information management, privacy, scalability.

I. INTRODUCTION

Over the last century, healthcare data management has been revolutionized by a wide range of hardware, software, and networking technologies all of which aim to improve the tracking of diseases and their causes, medical treatment, the quality of medical care and drugs, and to establish world-wide prevention plans for chronic diseases. The initial paper-based records have now transitioned to Electronic Heath Records (EHRs) [1]. EHRs need to be frequently distributed and shared among different hospitals, patients, clinics, pharmacists, medical insurance providers, medical drug manufacturers, researchers, and government to provide a holistic view of a patient's medical history in order to provide accurate and timely patient care. The distribution of EHRs becomes a time consuming and expensive process when we use the traditional client-server healthcare data management system where each hospital/clinic maintains its own database of patients' medical records. A patient's treatment is further delayed if the patient moves from one hospital to another hospital across different regions or countries. Moreover, most of the time a patient has to repeat several laboratory and radiology tests. Cloud-based health data management systems [2]–[4] have been introduced in the past to address the issues of scalability, real-time data access, single point of failure, privacy and security prevailing in the client-server-based system. The patients' medical data from different hospitals are stored in a remote cloud storage making it easily accessible by patients

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

and healthcare providers. However, this requires the patients and the hospitals to either encrypt the sensitive and private patients' medical data before uploading it to the cloud or to trust in the cloud service provider. The former requires a large amount of memory-intensive computing which is not suitable for a hospital environment [5] whereas, the latter may be very difficult to employ because the patients are aware of the potential risks of their data being misused [6]. Moreover, the issue of a single point of failure along with data security and patient's privacy risk prevailing in the client-server-based system, still exist in cloud-based systems. According to the statistics provided by the Health Insurance Portability and Accountability Act (HIPAA), 13,236,569 medical records were breached in 2018 which were as twice as compared to 5,138,179 records breached in 2017 [7].

Blockchain technology [8] which uses a shared, immutable, and transparent ledger has a great potential to solve the issues of real-time data access, vulnerability, data fragmentation, lack of traceability, security, and privacy which exist in the current client-server architecture. Each transaction in blockchain is timestamped and a block of several timestamped transactions is created. A block is linked to a previous block by using a cryptographic technique to serve immutability. The concept of timestamping to provide immutability was first introduced in 1991, when researchers at Telcordia Technologies (formerly Bellcore) proposed a mechanism to produce an immutable record of linked documents [9]. In this mechanism, a document owner sends the hashed document along with the owner's identity to a server for timestamping. The server digitally signs the document with the current timestamp and then hashes the signed document, sender's identity, and the previous document's hash together. Consequently, any attempt to either back-date or forward-date the document's timestamp will be discovered. This timestamping mechanism was improved later to add multiple documents to a single timestamped block [10].

Blockchain became a widely known concept with the introduction of bitcoin, a digital currency. In 2008, Satoshi Nakamoto published a white paper in which he proposed a direct online peer-to-peer payment without relying on a third party such as a bank or financial institutions [11]. The paper solved the problem of double spending in digital currency by linking every transaction to the preceding one in a public ledger producing an immutable record of transactions.

Blockchain eliminates the need for a central database server that operates as an intermediate among the peers. The server is a single point of failure. If the server fails, the entire network will be affected. In addition, the network bandwidth usage of the server must be high in order to support high volumes of network traffic. Moreover, this centralized approach also has security concerns because the server contains all the sensitive information of the network participants. Blockchain addresses these aforementioned issues through its decentralized network architecture wherein all the network participants have equal influence over the network and share a copy of the transactions' data in form of a ledger. This replication of data gives local access to the data and also helps to improve fault tolerance when data on some of the nodes is corrupted or the nodes behave maliciously. Furthermore, the data stored in the blockchain is immutable, i.e., once the data is stored, it cannot be modified or deleted. Any modification or deletion of the data is quickly detected by the underlying blockchain mechanisms. Each block in the chain is hashed using the hash of the previous block thereby creating an immutable chain. These distinctive features of blockchain has triggered its wide adoption for healthcare data management. However, most of the research for blockchain in the healthcare domain focused on the Bitcoin blockchain which suffers from several drawbacks such as high energy consumption [12], [13], poor scalability and low transaction throughput [14]. Such shortcomings must to be considered while developing a healthcare data management system underpinned by the blockchain technology.

We summarize the contributions of this work as follows:

- **We propose a lightweight blockchain architecture for healthcare data management that incurs low computation and communication overheads as compared to the traditional Bitcoin network. We achieve this by dividing the network participants into demographic clusters and maintaining one copy of the ledger per cluster. Since our proposed architecture does not fork the transactions on all the nodes as in Bitcoin and only the cluster heads maintain a copy of the ledger, we use the terminology 'lightweight'. to characterize the proposed architecture. In this architecture, we introduce the concept of canals that allow secure and confidential transactions within a group of network members.**
- **We propose a solution to avoid forking (which is prevalent in the Bitcoin network) by using a Head Blockchain Manager (HBCM) to generate blocks and handle the transactions.**
- **We evaluate the performance of our proposed blockchain architecture and compare it with the Bitcoin network. Our experimental results demonstrate that our proposed architecture outperforms the Bitcoin network in terms of the amount of network traffic generated and the time it takes to process data replication and ledger update.**

The rest of the paper is organized as follows. Section II presents an overview of related works. We describe the basic concepts of the blockchain technology in Section III. Section IV presents our proposed blockchain-based healthcare data management architecture. We describe transaction handling in Section V. Section VI presents a performance evaluation and analysis of our proposed architecture. Finally, Section VII concludes the paper.

II. RELATED WORKS

A. TRADITIONAL HEALTHCARE DATA MANAGEMENT

Healthcare data management is an important process where the patient's medical record is stored and managed to provide: improved care, efficient tracking of diseases and their causes, medical data for research and the development of effective medical drugs, and an efficient prevention plan. Currently, EHRs are widely used by hospitals and health providers to manage patients' medical data using a client-server architecture [15]–[21]. But in this type of healthcare data management system, the hospitals are the primary custodians of the data. This makes it difficult for healthcare professionals to make a precise disease prognosis or diagnosis when required and makes it difficult for patients to have a cohesive view of their medical history because their medical data is most likely to be stored at different hospitals and clinics. In order to allow a patient to track his/her own medical data from different organizations, several cloud-based healthcare data management systems [2]–[4] have been developed by academia and industry in the past few years. However, in these systems, a patient stores important health records in a cloud-based centralized database which suffers from a single point of failure making the system prone to errors, cyberattacks, and data loss. Consequently, the current client-server and/or cloud-based medical data management systems suffer from the issues of system vulnerability, data fragmentation, lack of accountability, security and privacy as we have mentioned previously.

B. BLOCKCHAIN-BASED HEALTHCARE DATA MANAGEMENT

Blockchain technology is one of the latest advances in information technology wherein the network participants can record transactions and immediately share them with other participants connected to the blockchain. Several research efforts used the blockchain to address the shortcomings of current EHR systems. Most of these works [22]–[26] have used blockchain to address security and privacy concerns of medical records by storing the hash of the cloud data in the blockchain. However, in these works, the system is vulnerable to a single point of failure because of the cloud server. In addition, the approach does not solve the problem of privacy of patients' medical records when they are stored in centralized cloud database. To address the issue of single point of failure many research works propose the use of blockchain to store the medical data in a distributed ledger. Most of these works either propose a new data encryption/decryption techniques [27]–[29], or a new digital signature scheme [30], or a secure data communication method [31], [32] or a key generator mechanism [33] which is used by the blockchain for medical data.

Very few works [34]–[40] have proposed health information systems using blockchain for sharing of patients' medical records among different hospitals. The authors of [34] propose a blockchain-based data sharing application, MedRec, that integrates with doctors' current data storage systems thereby facilitating interoperability. The application allows doctors to share patients' medical records on the blockchain. The authors of [35] and [36] propose a framework for accessing medical data using smart contracts in an Ethereum-based blockchain network. However, [34] uses patients' medical data as an incentive for mining which puts the patient's privacy at high risk. In addition, the blockchain network used in [34]–[36] is based on the Proof of Work (PoW) consensus mechanism which consumes a lot of energy [12], [13] and suffers from performance issues [14]. Moreover, these works require cryptocurrency tokens in order to initiate transactions necessary for data upload and data retrieval. The authors of [37] propose a blockchain-based medical data sharing system, MedBlock, that allows efficient access and retrieval of EHRs using low energy consuming consensus Practical Byzantine Fault Tolerance (PBFT) as compared to PoW [41] for a permissioned network. These works [34]–[37] do not allow patients to disseminate their medical conditions and lifestyles data to the blockchain network that would aid medical professionals for better prognosis/diagnosis and follow-up. In contrast, the authors of [38], [39] propose the use of blockchain to share patients' medical data. The patients upload the medical data while retaining the primary stewardship. However, these works only allow the medical professionals to view the patients' medical records and do not allow these professionals to disseminate patient's medical data (such as diagnosis, laboratory and radiology results, treatment, and vaccinations) to the network. The authors of [40] propose a health information system by using blockchain to share medical data among different patients and hospitals. This work allows both patients and hospitals to upload the patients' medical data to the distributed ledger and therefore provides a complete view of a patient's medical data history. However, the proposed framework uses a mining-based consensus approach which consumes a high amount of energy to generate a block. In contrast, our proposed architecture is highly scalable with high performance because it divides the blockchain network into clusters with a Blockchain Manager (BCM) per cluster maintaining one copy of the ledger and allow dissemination of patient's medical and personal health data. Moreover, our architecture uses canal that enables collaborating group within a network to perform confidential transactions maintaining a ledger accessible to the canal members. Energy-efficiency in our architecture is achieved by using the PBFT consensus mechanism. Table 1 outlines the strengths and the weaknesses of recent works on blockchain-based healthcare data management systems.

III. BLOCKCHAIN OVERVIEW

Blockchain is an immutable record of transactions stored in a ledger distributed among the participants to ensure decentralized and secure transactions. The transactions are grouped in a block and the block is linked to the chain. We define blockchain technology as a combination of three existing technologies which include distributed ledger [42], consensus protocols, and cryptography [43]. It is worth noting that

TABLE 1. Strengths and weaknesses of works on blockchain-based healthcare data management systems.

Work	Scalability	Energy-efficiency	Throughput	Privacy	Dissemination of personal health data	Dissemination of medical data	Remarks
[34]	✗	✗	Not mentioned	✗	✗	✓	The system does not scale and it consumes a high amount of energy because it uses the PoW consensus mechanism.
[35]	✓	✗	Not mentioned	✓	✗	✓	The system consumes a high amount of energy because it uses the PoW consensus mechanism. However, it is scalable because it stores only the hashes of the records in the blockchain while keeping the records in a database, and uses few nodes of the blockchain for transaction validation. Privacy is achieved by defining records sharing conditions using smart contracts.
[36]	✗	✗	Not mentioned	✗	✗	✓	The system does not scale and it consumes a high amount of energy because it uses the PoW consensus mechanism.
[37]	✓	✓	Not mentioned	✓	✗	✓	The system scales, consumes less energy and has low latency because it uses the PBFT consensus mechanism. Privacy is achieved by using a signature-based access control scheme.
[38]	Not mentioned	✗	Not mentioned	Not mentioned	✓	✗	The system consumes a high amount of energy because it uses the PoW consensus mechanism.
[39]	Not mentioned	Not mentioned	Not mentioned	✓	✓	✗	The system achieves privacy by using the proposed purpose-centric access control model.
[40]	✗	✓	Not mentioned	✓	✓	✓	The system does not scale because it uses the mining-based consensus mechanism. However, the system consumes less amount of energy compared to the Bitcoin network by reducing the number of miners. Privacy is achieved by using the proposed access control scheme.
Our proposed scheme	✓	✓	✓	✓	✓	✓	The system scales and is efficient compared to the Bitcoin network because of the division of the network into clusters with one Blockchain Manager (BCM) per cluster maintaining one copy of the ledger. We use the PBFT consensus mechanism that has low energy consumption. Privacy is achieved by using trusted canals to perform confidential transactions within a group of network participants.

Scalability: ✓→ the system scales when the number of nodes increases and ✗→ the system does not scale.
Energy-efficiency: ✓→ the system consumes low energy and ✗→ the system consumes high energy.
Privacy: ✓→ the system ensures data privacy and ✗→ the system does not ensure data privacy.
Personal health data: medical sensors and lifestyle data.
Medical data: the medical records that are maintained by the health medical providers, for example, diagnosis, treatment, laboratory and radiology results, vaccinations, and allergies.

these technologies are not new, but the application of these technologies combined together makes blockchain a new technology. We formulate the blockchain as,

$$Blockchain = \int (DL, CP, C) \tag{1}$$

We describe the parameters of this equation below.

- A Distributed Ledger (DL) is used to eliminate the need of a trusted third party in digital relationships and reduce the risk of a single point of failure. Blockchain uses a peer-to-peer network, where each network node holds a synchronized copy of the ledger. If a node

fails or behaves maliciously, the original data can still be retrieved from the other nodes, which is not possible if the data is stored by a central administrative authority. Thus, blockchain improves fault tolerance.

- Consensus Protocols (CPs) are used by blockchain participants to agree on a single state to update the ledger. The higher the number of nodes in the network verifying a state change, the more secure is the network. Blockchain uses a consensus protocol to validate the transactions, create a new block and append the created block onto the chain.
- Cryptography (C) technology is used to create a secure digital identity for every participant and validate transactions on the blockchain network . This is achieved by using a pair of public and private keys held by the participant.

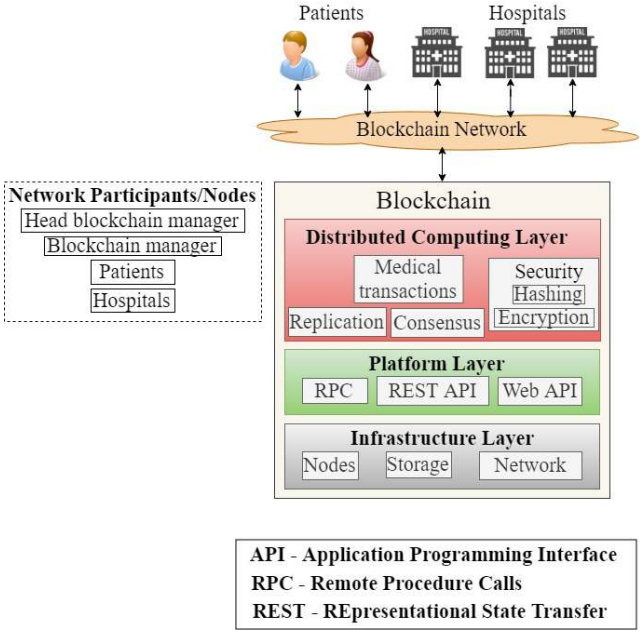

FIGURE 1. Overview of blockchain.

Figure 1 shows an overview of blockchain technology. We divide the blockchain architecture in three layers: infrastructure layer, platform layer, and distributed computing layer. The infrastructure layer consists of all the hardware components required to execute the blockchain. It includes nodes which are the network participants. A node can perform one or more of the following tasks: 1) initiate transactions, 2) validate transactions and blocks, 3) generate blocks, and 4) maintain a copy of the ledger. A network participant may have different roles such as client (patient and hospital), blockchain manager and head blockchain manager. The infrastructure layer also consists of the storage component which stores the ledger of the transaction records in the network. The other component of the infrastructure layer is the network facilities required for the communication within a blockchain or between different blockchains. The platform layer contains the facilities for invoking Remote Procedure Calls (RPCs) [44], web Application Programming Interface (API) [45], and REpresentational State Transfer (REST) API's [46] for the communication between the client and the blockchain network.

The distributed computing layer in the blockchain architecture ensures local access to data, fault tolerance, immutability, privacy, authenticity, and security of the transaction data. The ledger of the transaction records is replicated on distributed nodes connected in a peer-to-peer network serving for fault tolerance as well as immutability. Immutability is the blockchain property that does not allow modification of transaction records once updated in the ledger. The blockchain network uses a consensus algorithm to reach an agreement regarding the order of the transactions in the network, the update of the ledger, and decision about the next block generated. In addition, the distributed computing layer is responsible for user authentication via an encryption technique [47] and data privacy by using a hashing technique [48].

The blockchain architecture consists of different layers which have the following characteristics.

- *Decentralization:* Blockchain enables the sharing of a database directly in a distributed ledger without any intermediary entity. The transactions are processed and stored by the network nodes. The ledger is updated by the nodes after a consensus has been reached.
- *Transparency:* The transaction data replicated on the network nodes in the blockchain network is recorded as a chain of all the transactions linked together going all the way to the first transaction. Changes to the network are publicly visible making the network highly transparent and secure.
- *Immutability:* The transactions in the blockchain are stored in blocks. Each block in the chain is linked to the previous block using a cryptographic hash function. Any attempt to modify the content of a block will affect the subsequent blocks in the chain. Consequently, a malicious attacker requires to computationally change all the succeeding blocks in the chain to modify a particular block. This becomes difficult because the copy of the chained blocks is replicated over multiple nodes.
- *Traceability:* The distributed, transparent nature of the blockchain technology makes it easier to trace complex transaction events, such as in a supply chain [49]. Each update in the state of the asset can be traced back to its origin. This helps in making the blockchain network more secure, efficient, and transparent.
- *Trustless:* Blockchain enables the transaction of assets between unknown parties who do not trust each other. Distributing the ledger across several nodes in the network and updating this ledger via consensus ensures the validity of transactions in an untrusted environment.

A blockchain network is either public-centric or private-centric. Public blockchain, often referred to as permissionless blockchain, allows anyone to join the blockchain network without permission [50]. The user can join the network as

TABLE 2. Comparison between public and private blockchain networks.

	Public	Private
Network join permission	Open	Restricted/Authorized
Transaction visibility	All members	Selected authorized members
Participants of consensus process	All block generators	Selected nodes
Trust in the network	Not Required	Required
Data privacy	Low	High
Throughput	Low	High

a simple node, a validating node or a mining node. A simple node in the network can just send and receive transactions and does not store a copy of the ledger, neither validates a transaction, whereas a full node does. A Mining node is a full node with the capability of mining, i.e., the process of generating a new block. This type of network typically offers incentive for the users to participate in the consensus to encourage more participants to join the network. The identity of a network participant is kept pseudo-anonymous [51] by using a public key where the participant's real identity is kept unknown and recognized by a pseudo name. However, the transaction data is kept public over the network leading to the issue of data privacy [52]. Public networks are not highly scalable, and they also suffer from the issue of low throughput because the transaction validation and processing are done by a high number of participants spread across the world. Private blockchain, also called permissioned, is an invite-only network from an authentication authority [50]. Private blockchains have high transaction throughput and are more scalable than the public blockchains. The network involves access control rights for ledger query and updates. Table 2 shows the comparison between the public and the private blockchain networks.

A general blockchain network involves transactions, blocks, merkle tree root hash, previous block's hash, timestamp, block version, nodes, mining and genesis block.

- *Transaction:* A process that changes the state of the blockchain ledger. Depending on the application, the transaction can be the transfer of any valuable asset or the execution of a smart contract.
- *Block:* It consists of a block header and a block data. The header consists of the block's metadata information such as the merkle tree root hash, the previous block's hash, the time stamp, and the block version whereas the data consists of a set of valid transactions.
- *Merkle Tree Root Hash:* All the transactions in the block are hashed individually using a hashing algorithm. These hash values are then combined pairwise and are hashed again until a single hash value is obtained. This value is known as the merkle tree root hash value.
- *Block Hash:* It is the unique identifier of a particular block and is obtained by hashing the block header twice.
- *Previous Block's Hash:* It is the hash of the block preceding the current block in the chain. The preceding block is known as the parent of the current block.
- *Timestamp:* It indicates the time when the block was created.
- *Block Version:* It indicates the version of the validation rules followed by the blockchain network.
- *Nodes:* A typical blockchain network has three different type of nodes namely, simple node, full node and mining node.
- *Mining:* It is the process of adding the valid transactions in a block and broadcasting that block to the network.
- *Genesis Block:* This is the first block in the blockchain network, and it does not have any parent block. All the following blocks in the chain are linked to the genesis block. The genesis block generally includes the configuration for the network characteristics, consensus protocol to be used, access control rights, hashing function, block generation interval, and block size.

The mostly widely used blockchain is the public Bitcoin network wherein each user is associated with a pair of private and public keys. To propose a transaction, and to get it validated and added in the block, a user requires a pair of public and a private keys for transaction authentication and validation. The private key belonging to a user is only known by that user while the public key is known to everyone in the network. While creating a new transaction, the user first hashes the transaction data using a hashing function. The hashed data is encrypted using the user's private key and is then broadcasted along with the transaction data to the network. Each validator in the network validates the transaction to ensure the authenticity of the proposing node, integrity of the transaction data, and whether the node can perform that transaction. In order to do so, each full node decrypts the encrypted transaction data using the public key of the proposing node and obtains the hash value. Then the full node hashes the transaction data to generate the hash value and then matches it with the decrypted hash value. This ensures that the transaction data has not been tempered with and it is from an authentic user. The transaction, once valid, is broadcasted to the miners in the network. The miner (selected according to the consensus protocol) verifies the valid transactions and group them in a block. The transactions are grouped in such a way that the block size does not exceed a predetermined threshold.

In a Bitcoin network, all the miners compete against each other to mine a new block and the network agrees on which

miner's block to add in the chain by using a consensus protocol. The consensus protocol used in the Bitcoin is Proof of Work (PoW) [53], where each miner competes to find a nonce value such that the hash of the current block has a predefined number of leading consecutive zeros and is below a threshold limit. The nonce is a variable whose value is adjusted by a brute-force method in order to produce the desired block hash. In PoW, each miner accumulates valid transactions in a block and computes the merkle root hash value. The root hash, the previous block's hash, the timestamp, the block version, and the nonce are all input to a hash function to compute the current block's hash value. The nonce value is brute-forced until the desired block hash is found. Once the nonce generating the threshold hash has been found, the miner broadcasts the block to the network. All the other miners stop the mining process and verify the validity of the proposed block. This is done by checking if: the block number is correct, all the transactions in the block are valid, the block has a valid parent hash (by verifying the previous block's hash), and the block's hash is as required (by recalculating the block's hash using the provided nonce value in the block). Once valid, the block is appended to the chain and the miners start mining the next block. The previous block's hash value is used as an input to compute the current block's hash so as to ensures the immutability of the blockchain ledger. For instance, if any malicious user tries to modify a transaction data in a block somewhere in the past, the hash value of that block will change. This hash value will not match the stored value in the parent block hash field in the next block which will reveal the malicious attack. To modify any transaction, the user has to modify all the blocks in the chain as well as all the copies of the ledger distributed among different the users which is computationally impossible. However, PoW consumes a lot of energy and has low transaction throughput. In 2017, the bitcoin mining used around 30.14 Terrawatt hours (TWh) of energy, which is equivalent to the energy usage of Ireland in a year [54]. The annual energy consumption as of 25 August 2019 was 73.121 TWh [13]. This high amount of energy consumption of the Bitcoin network has an adverse effect of the environment. According to a research, the annual carbon dioxide emissions by the bitcoin network are as high as 22.9 million metric tons, almost equivalent to the amount produced by the countries like Sri Lanka and Jordan [55].

IV. PROPOSED BLOCKCHAIN-BASED HEALTHCARE DATA MANAGEMENT ARCHITECTURE

Figure 2 shows our proposed architecture for blockchain-based healthcare data management. It shows the different architectural components along with their roles and relationships. This section describes the roles of each component. We use the permissioned blockchain network over the permission-less because of the following issues associated with the latter: 1) unauthorized network participation leading to impersonation of network members, 2) clear transaction data in the ledger accessible to each network participants revealing sensitive patient's data, 3) slow network throughput hindering real-time patient's treatment, and 4) the need for paying transaction execution fees and mining rewards limiting the usability of the network. Next, we briefly describe the different components of the proposed architecture.

A. HEAD BLOCKCHAIN MANAGER (HBCM)

The Head Blockchain Manager (HBCM) is the main authority of the blockchain that regulates the network. HBCM acts as a Certificate Authority (CA) providing valid digital identity to the participants to join the network. Moreover, HBCM receives the transactions from the clients and generates the blocks. Thus, the role of a miner in our architecture is done by the HBCM. Compared to the Bitcoin network, where there are multiple miners competing against each other to generate a block, our proposed architecture uses a single HBCM for this purpose in order to reduce computational overhead and to address the issue of high energy consumption associated with Bitcoin. The HBCM maintains a Ledger (L) of all the transactions that are submitted by the client (valid and invalid). The role of HBCM can be played by either the national ministry of health or some well-known medical organizations and society such as the Healthcare Information and Management Systems Society (HIMSS) [56]. In order to ensure single point of failure and data availability our architecture uses two HBCM (leader and follower) as shown in Figure 2. The leader HBCM generates the blocks while the follower receives the blocks from the leader and replicates them. The selection of leader is done by using the method proposed in [57], i.e., based on the number of votes.

In a public blockchain network such as Bitcoin where there is no restriction for a node to join the network and participate in the process of mining, there exists a high number of miners throughout the network. It is possible that two miners, separated geographically, mines a valid block simultaneously and broadcasts it to the network. Depending on the location and network connectivity some of the nodes in the network may receive a block (let us call it block A) from one miner and other nodes receive a block (let us call it block B) from another miner. All the nodes in the network maintain a copy of the blockchain ledger (let us call it the main chain). A node, which receives block A, validates it and appends it to its copy of the main chain and a node, which receives block B, validates it and appends it to its copy of the main chain. When a node having block A in its main chain receives block B, the node verifies the validity of block B. However, both blocks A and B have the same parent block. Consequently, the node initiates a new chain (let us call it the secondary chain) separate from the main chain and appends the block B in that secondary chain. Similarly, a node having block B in its main chain appends block A in its secondary chain after it has been received and validated. In the blockchain, when the chain is divided into two parts in this way, it is known as forking [58]. In order to manage forking, the blockchain network uses the rule of the longest chain. This rule states that if the next block to be mined in

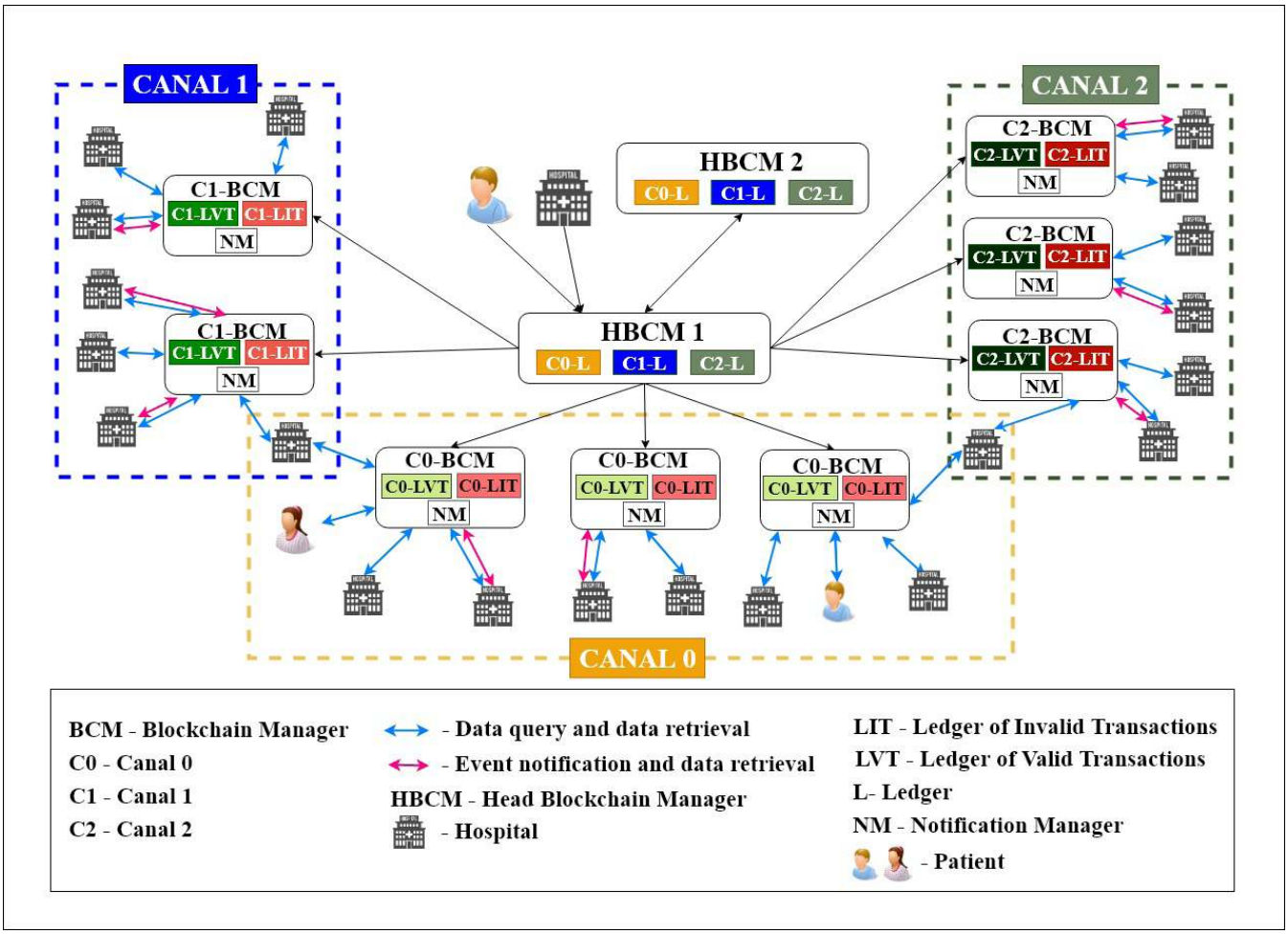

FIGURE 2. Proposed architecture of blockchain-based healthcare data management system.

the network will have block A as a parent block, then the blockchain with block A in the main chain will be considered valid as it becomes long compared to the one with block B in main chain. Similarly, if the new block contains block B as the parent block, the blockchain with block B in the main chain will be considered valid. The blocks in the secondary chain are then removed and the chain continues to grow on the main chain. Generally, the blockchain fork is managed within one block [58]. Thus, to handle transactions in the forked block, all the transactions in a block in the blockchain are executed after certain number of blocks are appended to the longest chain. For instance, in a bitcoin network, it takes 6 block confirmations for a transaction to get executed [59]. However, in the healthcare domain where the delay in accessing patient's critical medical data can trigger life threatening situations, forking is not acceptable. This issue is addressed in our architecture because all the blocks are generated, and the transactions are handled by the HBCM.

B. BLOCKCHAIN MANAGER (BCM)

In a blockchain network similar to that of Bitcoin all the hospitals maintain a replicated copy of the ledger. The replication of the ledger on all the nodes increases the computational and network overheads leading to high energy consumption, low transaction throughput and low scalability. To address this issue, our architecture creates clusters of hospitals where the ledger is maintained by only one cluster node and can be queried by the others. Consequently, the grouping of the nodes into clusters in our proposed architecture eliminates the need of data communication and replication on all the blockchain network nodes thereby reducing the computational and communication overheads compared to the Bitcoin network. The hospitals are divided into clusters based on their demography to avoid unnecessary communication delays which cause higher latency. Within each cluster, one node is selected as the Blockchain Manager (BCM) which participates in the process of consensus and maintains a copy of the ledger. The selection of the BCM is done by using the method proposed in [60], i.e., in which a cluster node having the maximum number of incident edges is selected as the BCM. Each BCM in the network receives and verifies the block generated by the HBCM. It then constructs and maintains a Ledger of all the Valid Transactions (LVT) and a Ledger of all the Invalid Transactions (LIT) in the network.

These ledgers are replicated on all the BCMs in the network.

C. CANAL

Medical data is very sensitive, private, and important information. When we use a permission-less blockchain where the data is public and can be accessed by anyone, data privacy is a concern. One solution to address this issue is to use a permissioned network where only the authorized participants can view and access the data based on the access rights defined for the data. However, there can be situations where a sub-group of the authorized group needs to perform some confidential transactions. For instance, when a group of hospitals collaborate together for some confidential research or to track some illegal activities (such as false billing, duplicate medical claims, forged diagnostics, bribe, unwanted services, and data breaching). To facilitate this, our architecture allows collaborating network participants to create a sub-network, that we call a canal, to perform private transactions. The transactions in a canal are only visible to the participants of that sub-network, while the other network participants are unaware of the canal's transactions. This is done by creating a separate blockchain ledger for each canal. The network can support as many canals as possible. Each incoming transaction to the leader HBCM contains the information about the canal it belongs to. The HBCM places the transaction in the blockchain ledger corresponding to that canal. The blocks in that ledger are broadcasted only to the BCMs which are part of that specific canal. As shown in Figure 2, there exists three canals: canal 0, canal 1, and canal 2. Canal 0 involves the public blockchain ledger which can be viewed and accessed by all the network participants. On the other hand, the transactions in the canals 1 and 2 are confidential and can only be viewed and accessed by the respective canal members. Hospitals in each canal are divided into clusters with each cluster having a BCM.

D. LEDGER

The ledger in blockchain consists of all the transactions that have occurred in the blockchain network. These transactions are stored in blocks where each block is chained to its previous block by using a cryptography technique. In our architecture, the ledger is maintained by the HBCMs and all the BCMs in the network. The ledger of HBCMs contains all the valid and invalid transactions submitted to the network. The inclusion of invalid transactions in addition to valid transactions is for auditing purposes. In order to manage the canals, the HBCMs maintains a ledger for each canal (Figure 2. The BCMs of each canal receives the corresponding canal ledger from the HBCM. Each BCM verifies the transactions contained in the received block and separates the valid ones from the invalid ones into two different blocks. The block of valid transactions is appended to LVT and the block of invalid transactions is appended to LIT. Both ledgers exist in the same blockchain network replicated on all the BCMs in a canal based on a consensus as shown in Figure 2. The ledger of invalid transactions serves for auditing purposes, particularly in the case of double spending. The hospitals in the cluster (other than the BCMs) and the patients can then query the BCMs to view and access the medical data. The queried patient data is then stored in a local database by the hospitals. The patient query will be processed by the BCM nearest to the patient in terms of physical location.

E. NOTIFICATION MANAGER (NM)

Each BCM in our proposed architecture consists of a Notification Manager (NM) to handle events' notifications in the network. An event is defined as the successful execution of a transaction. The patient and hospitals can choose to receive notifications for different events. We classify the events into patient events and hospital events. The patient events include: 1) an update of the patient's record by the hospital(s) where the patient is receiving treatment, 2) a data access query of the patient's record by a hospital seeking research data in the blockchain network, and 3) a successful data update request by a patient. The hospital events include: 1) an update of medical and lifestyle data of the patient, 2) a medical data access request of a patient by another hospital to a hospital maintaining the patient's medical database, 3) and the successful data update request by a hospital. The patients and hospitals that opted for notifications will be notified by the NM about the occurrence of a patient event and hospital event respectively. The subscription for events' notifications is handled by the NM. For a hospital to subscribe for notifications via the NM, the hospital should belong to the cluster of the corresponding BCM. Similarly, for a patient to subscribe to the NM notifications, he/she should be registered with the hospital that belongs to the cluster of the corresponding BCM.

F. CONSENSUS

The blockchain network does not have trusted third parties and uses consensus protocols to reach an agreement by the network participants that do not trust each other. The consensus helps to agree on the order of transactions in the network, the update of the ledger, and the network characteristics and policies. The consensus protocol used by the Bitcoin network is PoW which suffers from the issue of high energy consumption and low transaction throughput caused by the complexity of computation to select a minder. In PoW a miner would refrain from mining invalid block because of the invested computational mining power. However, many miners now form groups to distribute the computation and increase the chances for mining of next block in the chain. This group of miners is known as a mining pool. Each miner in a pool uses its computing capacity for computing PoW. If a mining pool owns more than 50% of the network's computing power, it is likely that this group would be able to prevent the validation of transactions. This is known as the problem of 51% attack in the Bitcoin network [61]. To address the aforementioned issues in PoW, our proposed blockchain architecture uses

the Practical Byzantine Fault Tolerance (PBFT) consensus protocol.

PBFT was proposed by Castro et al. in 1999 [62], where all the network nodes communicate with each other with the goal of reaching an agreement by assuming that all honest nodes will have the same exact copy of the ledger. In our architecture, the leader HBCM sends the generated block (that includes both valid and invalid transactions) to all the BCMs in a canal. Each BCM verifies all the transactions in the received block and will group the valid and invalid transactions into two different blocks. For each block, the BCM calculates the merkle tree root hash and concatenate this root hash value with the previous block's hash, block number, timestamp and block version and sends it to a hash function to calculate the block's hash. This is done for both the block of valid transactions and the block of invalid transactions. Each BCM then broadcasts these blocks' hash values to other BCMs in the same canal. A BCM waits until it receives the same hash from f + 1 or two-thirds of the BCMs in the network, where f represents the number of faulty nodes. Once received, the BCM updates its LVT with the block of valid transactions and its LIT with the block of invalid transactions. For the PBFT protocol to function correctly, the number of malicious or crashed BCMs must not be equal or greater than $\frac{n}{3}$ out of the total n BCMs in the same canal at a given time. Thus, the higher the number of nodes in the network, the more unlikely it is for more than one third nodes to be malicious. Consequently, the more secure is the network.

G. DATA REPLICATION

Replication in a distributed system is a mechanism that enables geographically distributed nodes to access a shared replicated data ledger. In our proposed architecture, each BCM in a canal shares the replicated ledger. Ideally, the copy of the ledger should be consistent across the different BCMs and should always be available. However, in a distributed system where network partition may occur, data messages could be delayed or lost. Consequently, ensuring high consistency and high availability at the same time is a challenge and a trade-off [63] needs to be achieved. Based on this trade-off between consistency and data availability along with network performance in terms of latency and transaction throughput, the data replication strategies in large-scale distributed systems can be divided into five different consistency models: strong consistency, consistent prefix, bounded staleness, monotonic reads, and eventual consistency [64]. We explain these consistency models in the context of the blockchain technology. The execution of a transaction involves 'write' operations to the ledger whereas querying an executed transaction involves 'read' operations. Table 3 compares different consistency models used for replication strategy in distributed systems.

- *Strong Consistency:* In this strategy, all the nodes are synchronized after each execution operation in the data ledger. The read operation in this type of model always guarantees to get the latest updated status of a ledger from all the nodes. A node trying to get the status of a transaction will wait as long as there is a transaction being processed. This increases the latency of the network in particular in a dynamic network where the transactions arrival rate is high and/or the network bandwidth and latency are important issues.
- *Consistent Prefix:* The transactions can be executed, and the nodes will be synchronized later. In this strategy, the nodes can access a consistent prefix of the ledger. All the nodes are synchronized on the prefix ledger. However, this strategy does not guarantee the same order of transactions in all the nodes of the network.
- *Bounded Staleness:* This strategy allows a node to read a transaction state before a complete ledger is updated on all the nodes. Updates of the nodes' ledger is done in an asynchronous way. The read operation will get transaction status updates (not necessarily the entire updated ledger)from each node, where the last updated blocks were done before a specified period of time T. This technique does not allow a node to update its own ledger and to have the same status of a full ledger.
- *Monotonic Reads:* In this strategy, each transaction is prefixed with a timestamp or index. This ensures that all nodes have the same ledger with the same order of transactions.
- *Eventual Consistency:* It is the weakest of all the above consistency models. It does not guarantee the order of the transactions to be the same at each node. The read operation of an executed transaction in this model results in any arbitrary data state. This is because of the delay in the ledger update. However, the model assumes that the ledger will eventually be consistent.

TABLE 3. Comparison of consistency models for replication strategies in distributed systems.

Consistency model	Consistency	Availability	Latency	Throughput
Strong consistency	Very high	Very low	Very high	Very low
Consistent prefix	Low	High	Low	Moderate
Bounded staleness	High	Low	High	Low
Monotonic reads	Moderate	High	Moderate	Moderate
Eventual consistency	Very low	High	Low	Very high

For a blockchain network that tolerates partitioning in large-scale distributed systems and requires all the nodes to maintain the ledger to agree on the entire chain in a synchronized manner, the implementation of strong consistency will be impossible. Thus, an appropriate consistency strategy for the blockchain network should allow the nodes in the network to agree on the current chain (i.e., consistent prefix) except for a potentially small number of unconfirmed blocks [65]. However, having a consistent prefix is not enough for blockchain. This is because the blocks are not updated during synchronization on all the nodes. A node might be having a copy of the ledger with M blocks at time T in the network while another node has a copy of the ledger with N blocks (N < M)

at time T' (T' > T). Consequently, a node requesting a recent block might get block N to update its own ledger believing it is a more recent block update than block M. Therefore, blockchain implements the monotonic reads replication strategy as well. Based on the discussion, our proposed blockchain architecture uses a combination of two replication strategies: consistent prefix and monotonic reads; that is referred as Monotonic Prefix Consistency (MPC) [66].

V. TRANSACTION HANDLING

In this section, we discuss how transactions are handled in our proposed architecture. First, we discuss how the client's (patient's or hospital's) data is pushed to the blockchain network and updated on the ledger. Second, we explain how a query transaction is launched by a hospital or a patient to retrieve a medical record from the ledger.

A. UPDATE

To update a medical or health record, a client first prepares the transaction proposal to send it to the leader HBCM. A transaction proposal includes the patient ID, the hospital ID of the patient to which the data belongs to, the medical data, the canal to which the data belongs, and the timestamp. The transaction proposal is hashed using a hashing function and the hashed value is then encrypted using the client's private key. The encrypted value acts as the digital signature for that transaction proposal. The transaction proposal and the corresponding digital signature are broadcasted to the leader HBCM. The leader accumulates all the transactions received from different clients and creates a block of transactions. Each block contains transactions belonging to the same canal. Once the block threshold limit is reached, i.e., a permissible number of transactions is reached or the block is limited by the block size, the leader HBCM calculates the hash of the block (as described in Section IV-F) and broadcasts the block to the follower HBCM and all the BCMs in the canal where the block belongs to. The follower HBCM replicates the block on its ledger, while each BCM verifies the transactions in the block for its validity. The BCM decrypts the proposed encrypted transaction using the client's public key to obtain the hash value of the transaction. The successful decryption ensures the client's authenticity. The BCM then hashes the proposed transaction that was sent along with the signature and compares it with the hash value obtained by decrypting the digital signature. If both hash values match the data integrity is ensured. This process of ensuring client's authenticity and data integrity is known as transaction verification. The BCM separates the valid transactions from the invalid ones. A block of valid transactions and a block of invalid ones are created and the hash values for each block is calculated. Each BCM broadcasts their valid and invalid blocks' hashes to other BCMs in the canal. If a BCM receives the same hash values for the block from f + 1 (f is the number of faulty BCMs) or two-third of the other BCMs, then the ledger of that BCM is updated. The hospital and the patient corresponding to the hospital ID and the patient ID respectively in the transaction proposal are then notified about the transaction if they have opted for notifications. The hospital can then query its BCM for the data which is then stored in the hospital's local database upon retrieval. Figure 3 shows the transaction update flow diagram.

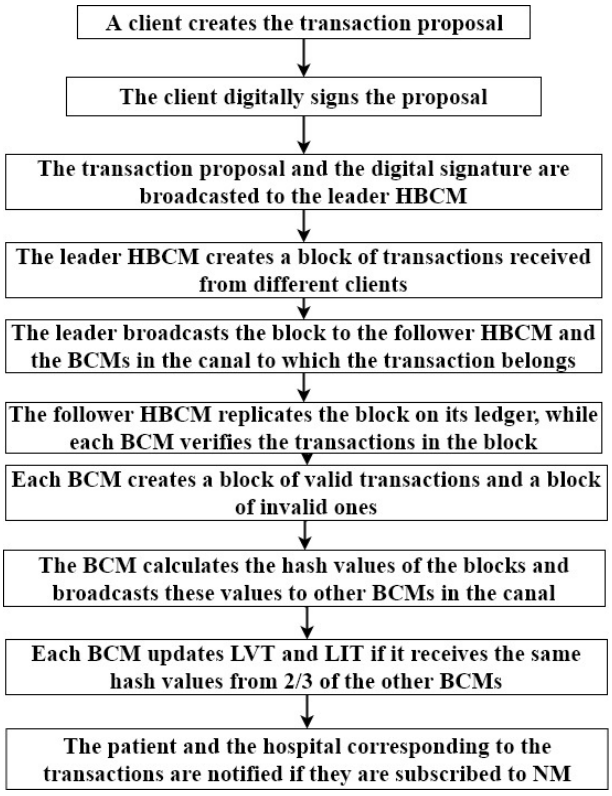

FIGURE 3. Data flow process for transaction update.

B. QUERY

The query transaction is executed in a similar way as the update transaction. The query by a hospital is responded by the BCM of the cluster to which that hospital belongs to. In contrast, the query of the patient is handled by the BCM which is nearest to the patient in terms of physical location to avoid communication delay. If that BCM is not available to respond the patient's query, the query is handled by the next nearest BCM.

VI. PERFORMANCE EVALUATION

In this section, we first compare the proposed blockchain architecture and the Bitcoin blockchain in terms of security and privacy considerations. Then, we simulate a blockchain network and evaluate the performance of the Bitcoin network and our proposed architecture. We analyze and compare the performance in terms of amount of data transferred and processing time required for data replication and ledger update. We also evaluate the impact of number of blocks and nodes on the performance.

A. SECURITY AND PRIVACY ANALYSIS

In this section we discuss some possible security and privacy threats for a blockchain network and explain how our

proposed architecture will handle these threats. It is believed that a malicious attacker can be the BCM, patients, hospitals or a person who is not part of the network. Attackers can impersonate a user identity, create fake blocks or transactions, hinder the communication, discard transactions, delete or modify a transaction data, or link a user's transaction to its identity. The main classes of threats which exist in the Bitcoin network can be classified into three categories [67]: 1) accessibility threats, 2) anonymity threats, and 3) authentication and access control threats. Accessibility threats cause problems for a blockchain participant to access his/her data, whereas anonymity threats link the user's transaction to his/her identity by analyzing the public blockchain transactions and other available information. The different types of accessibility threats include Denial of Service (DoS), data modification, blocks dropping, and appending of invalid blocks. Authentication and access control threats involve impersonation of a blockchain user in order to gain access to his/her data. Next, we discuss these treats and explain how our proposed architecture addresses them.

In a DoS attack, a malicious attacker sends many transactions repeatedly to a network participant to reduce its availability. In our architecture, all the transactions are processed by the HBCM to mitigate this type of attack. If the HBCM receives several unsuccessful transactions in the same canal, it can block that attacker node from accessing the network. The list of unsuccessful transactions is known in our architecture because the ledger of HBCM contains both valid and invalid transactions. In the data modification attack, an attacker tries to modify or delete the data of a particular user. The data of the BCMs cannot be modified due to the immutability property of the blockchain. If the attacker tries to modify or delete any data from the hospitals, then the data can be queried from the corresponding cluster's BCM.

In the blocks dropping attack, an attacker can take control of a BCM and then drops all the received blocks from the HBCM. Such an attack would be detected in our proposed architecture because the patients and hospitals would not be able to query the data. In this case, a new BCM can be selected. In the attack which involves appending invalid blocks, an attacker should have control over multiple BCMs simultaneously. However, as the BCMs are scattered in different canals in our architecture, it is not possible to attack multiple BCMs because of the secure canal. In the anonymity threat, an attacker may try to link a user's transactions to his/her identity based on the user's transaction history and other available public information. This linking of transactions with a user identity is not possible in our architecture because of two reasons. First, our proposed architecture uses a permissioned blockchain network as compared to Bitcoin's permission-less network where the transaction data is made public. Second, a user's transactions are scattered in different ledgers based on the canal used, which makes it impossible to gather all the information about user's transaction data. In the authentication and access control threat, an attacker can try to take control over the user's (patient's or hospital's) identity to access his/her data. In our architecture, the ledger is not stored at the user's node restricting the access to data only by query. Multiple queries from a user to BCM can be fielded to the HBCM which further investigates the issue.

B. PERFORMANCE ANALYSIS

In this section, we analyze and compare the performance of our proposed blockchain architecture with the Bitcoin network. We did this by simulating a Bitcoin network and our proposed blockchain architecture using NS3 simulator. The nodes in each of the simulated networks are connected using a random topology similar to that in a blockchain. In the simulated Bitcoin network, each node is connected to 10 other nodes, with a ledger replicated on all the nodes using broadcasting. However, in the simulated proposed architecture, the network nodes are divided into clusters with each cluster having a BCM. The ledger is then replicated on the BCMs. The selection of the number of clusters is based on the work by [68], according to which the optimal number of clusters in terms of energy consumption and performance for a group of 100 nodes is 10. Figure 4 shows the experimental setup for the Bitcoin and the proposed architecture networks. Similar to a blockchain network, we use blocks to maintain the record of transactions with each block having a hash value. The block size used in the simulation is 1 MB and the hashing algorithm used is SHA-256. We choose SHA-256 due to its popularity among the Bitcoin network. Table 4 shows the parameters we have used to simulate the blockchain networks, and the specifications of the server we employ to execute the simulation. We evaluate the performance of the Bitcoin network and our proposed architecture in terms of the amount of data transferred (in MB) over the network in two different scenarios: 1) when the number of blocks increases, and 2) when the number of nodes increases. In addition, we observe the total execution time for data replication and ledger update in both scenarios. We repeat each experiment 10 times and we calculate the average values.

In the first scenario, we keep the number of nodes constant at 100 and increase the number of blocks from 10 to 50 at an interval of 10. These 100 nodes are divided into 10 clusters with each cluster having a BCM to simulate our proposed network. To calculate the amount of data transferred for the Bitcoin network, 10 blocks of 1 MB each generated by a network node are broadcasted to the 99 other nodes of the network to update the ledger of each node. As each node in the network is connected to 10 other nodes, at most 2 hops are required for a block to reach a node. We then measure the total amount of data transfer during this broadcast. In our proposed architecture, the block is generated by the leader HBCM and is broadcasted to the follower HBCM and 10 BCMs in the network. We calculate the total amount of data transferred during this broadcast. In addition to the broadcast of the block in our architecture, each BCM broadcasts the hash calculated by that node to the remaining 9 BCMs in the network. The size of a block's hash value is 256 bytes due to the use of SHA-256 hashing algorithm. Consequently, the total

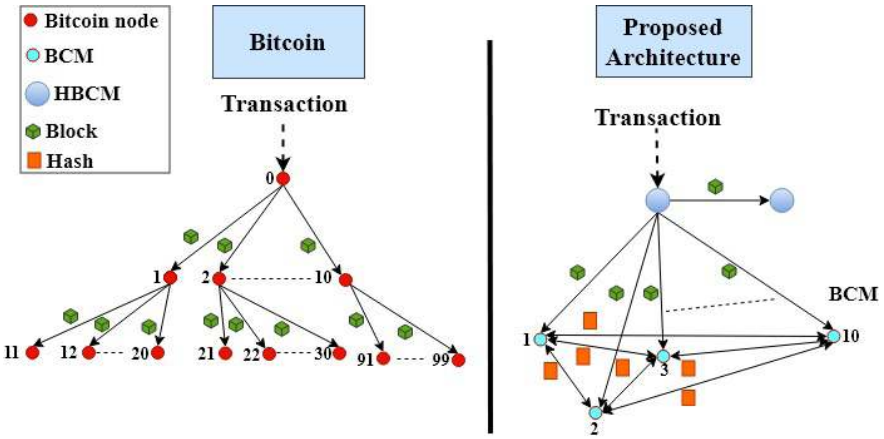

FIGURE 4. Experimental setup for the blockchain networks.

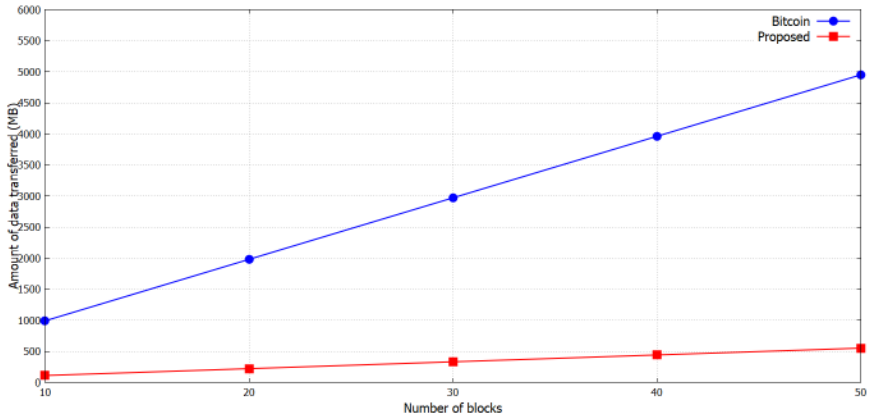

FIGURE 5. Amount of data transferred using Bitcoin and our proposed blockchain architecture when the number of blocks increases.

TABLE 4. Experimental environment and setup.

Server Specifications	Processor	Intel Xeon X5355 @ 2.66 GHz (8 cores)
	Memory	8GB RAM
	Storage	2 x 146GB
	Network interface	1GbE Intel 80003ES2LAN
	Operating system	CentOS release 6.5 - 64bit
Simulation parameters	Block size	1MB
	Block's hash size	256 bytes
	# Blocks	10, 20, 30, 40, and 50 (number of nodes and clusters constant at 100 and 10 respectively)
	# Nodes (clusters)	100 (10), 200 (20), 300 (30), 400 (40), and 500 (50) (number of blocks constant at 10)

amount of data transferred in our architecture is calculated by adding the data transferred during the broadcast of the block and the data transferred during the broadcast of the block's hash value. To calculate the processing time, we simulate the network in a similar way and initiate the broadcasts of blocks using the Bitcoin network and our architecture. We recorded the total processing time required by the network for the successful data replication and update of the ledger during the simulation. In the second scenario, we keep the number of blocks constant at 10 and increase the number of nodes from 100 to 500 at an interval of 100. We divide the nodes into clusters in the same way as in the first scenario (i.e., 10 clusters for 100 nodes, which means that 20 clusters for 200 nodes and so on). In addition, we calculate the amount of data transferred and the total processing time for the Bitcoin network and our proposed architecture the same way as in the first scenario.

Figure 5 shows the amount of data transferred (MB) for updating the ledger at each node in the Bitcoin network and at each BCM in our proposed architecture when the number of blocks increases. The amount of data linearly increases for both cases with the Bitcoin network using more data leading to higher computational and traffic overheads. On average the amount of data transferred using the Bitcoin network is 11 times higher than our proposed network. This improved perform is because of the clustering approach in our proposed

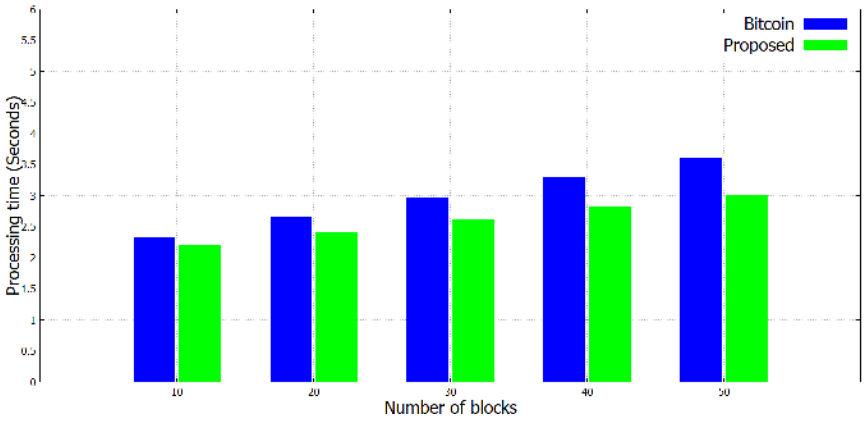

FIGURE 6. Processing time using Bitcoin and our proposed blockchain architecture when the number of blocks increases.

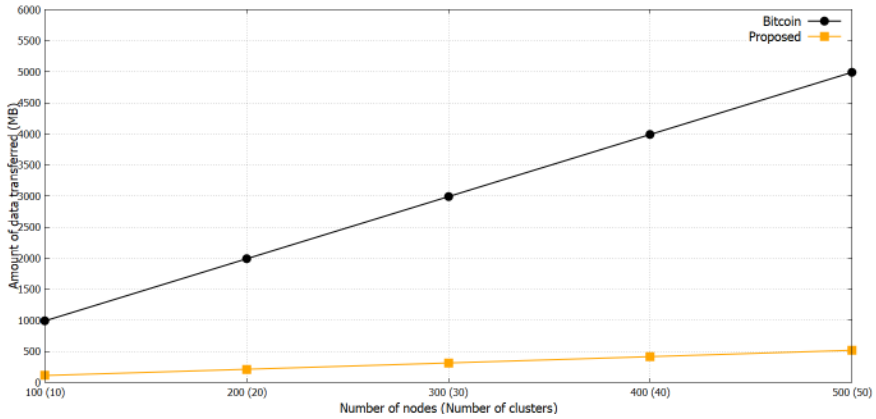

FIGURE 7. Amount of data transferred using Bitcoin and our proposed blockchain architecture when the number of nodes increases.

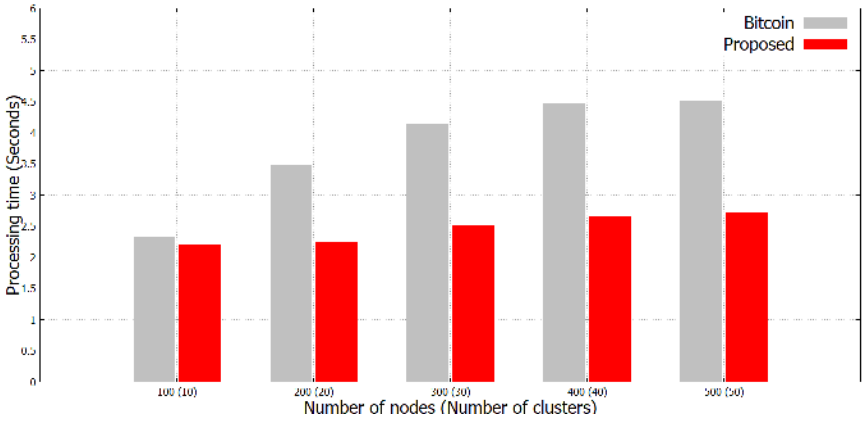

FIGURE 8. Processing time using Bitcoin and our proposed blockchain architecture when the number of nodes increases.

architecture which performs data replication only on BCMs as compared to Bitcoin which replicates data on all the nodes thereby requiring more broadcasts.

Figure 6 shows the processing time for the Bitcoin network and our proposed architecture when the number of blocks increases. It shows that our proposed architecture takes less time to replicate data and update the ledger compared to the Bitcoin network. This is because of the clustering approach in our architecture which eliminates data replication on all the nodes. On average when the number of

blocks increases, the Bitcoin network takes 1.13 times more time to replicate the data compared to our proposed architecture. The difference between the processing times with both approaches increases when the number of blocks in the network increases.

Figures 7 and 8 show the amount of data transferred (MB) and processing time respectively for updating the ledger at each node in the Bitcoin network and at each BCM in our proposed architecture and the when the number of nodes increases. They show that our proposed architecture outperforms the Bitcoin network both in terms of data transferred and processing time, demonstrating scalability of our architecture. On average when the number of nodes increases, the amount of data transferred using the Bitcoin network is 10 times higher than our proposed architecture. Moreover, our architecture yields an average improvement of 67% in processing time required for data replication and ledger update compared to the Bitcoin network.

VII. CONCLUSION

Healthcare data management has been gaining increasing attention in the last few years as it can provide more accurate, efficient, and cost-effective patient care. Blockchain technology has strong potential to improve the management of medical data because it can address issues such as single point of failure, data stewardship, system vulnerability, distributed information, and high security and privacy risks prevailing in the existing client-server and cloud-based approaches. However, most of the recent research efforts aimed at implementing blockchain in the healthcare domain have focused on the Bitcoin network. However, as we have mentioned previously, the Bitcoin network suffers from high energy consumption, low transaction throughput, limited scalability, and privacy and security threats. Consequently, there is a need for a more scalable and efficient blockchain architecture. In this paper, we have proposed a lightweight blockchain architecture for healthcare data management that has low computational and communication overhead as compared to the Bitcoin network. We replaced the energy consuming mining consensus protocol of the Bitcoin network with a scalable and an energy-efficient consensus protocol. Moreover, our architecture divides the nodes into clusters, with each cluster having a manager that maintains the ledger as compared to the Bitcoin network where all the nodes maintain the ledger. Consequently, our architecture reduces the computational and communication delay making it more scalable. Our architecture uses a HBCM which generates the blocks (replicated on the other BCMs) and orders the transactions. This approach where a single entity generates a block solves the issue of forking that is prevalent in the Bitcoin network which if it exists in the healthcare domain can trigger life threatening situations. We analyze the effectiveness of our proposed architecture in providing security and privacy by examining different threat models which exist in the Bitcoin network and we discussed how our architecture addresses them. We also simulate the blockchain network to evaluate and compare the performance of our architecture with the Bitcoin network in terms of amount of data transferred and network processing time for data replication and ledger update. Our performance results demonstrate that our proposed architecture generates $\frac{1}{11}$ of network traffic compared to Bitcoin when the number of blocks increases. Our ledger update is 1.13 times faster. Furthermore, results show that the network traffic is $\frac{1}{10}$ that of Bitcoin and has a speedup of 67% in ledger update when the number of nodes increases.

The replication of the ledger in Bitcoin ensures data security and privacy in a permissioned network where the participants do not trust each other. However, in healthcare domain where the hospitals and medical organizations form a trusted network, using Bitcoin would lead to scalability and energy issues as discussed. Our architecture uses the clustering technique to increase scalability and reduce energy consumption in healthcare domain while satisfying the security requirements. The results obtained with our proposed architecture demonstrate its efficiency and its attractiveness for potential adoption by medical organizations seeking to use blockchain technology for healthcare data management.

There has been an increasing use of medical sensors for remote and real-time health monitoring with the introduction of computing paradigm such as the Internet of Things (IoT) in the domain of smart health. However, these sensors generate and process a vast amount of private data related to a patient's health condition that has to be secured from cyberattacks. Introducing a blockchain network such as Bitcoin to healthcare IoT is not straightforward because of high resource requirements for PoW consensus, communication delays, and computational overheads. Fog computing can increase the performance of blockchain for smart health applications. However, the performance can be further improved by incorporating our proposed architecture using clustering techniques along with fog computing. For instance, the data coming from sensors can be processed by a mobile gateway fog node, acting as the head of a cluster of sensors. The fog node maintains a record of the sensors' data transactions. A cluster of multiple fog devices can be formed based on the patients registered with a particular hospital or based on the geographical location of the patients. The head of this cluster can then maintain a copy of the ledger. Currently, we are integrating IoT into our proposed architecture for smart and remote healthcare blockchain solution. Our future work will focus on a real implementation of the architecture in order to extensively evaluate its security, privacy and performance.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable comments which helped us improve the content, quality, and presentation of this paper.

REFERENCES

[1] E. Jamoom, N. Yang, and E. Hing, "Adoption of certified electronic health record systems and electronic information sharing in physician offices: United States, 2013 and 2014," U.S. Dept. Health Hum. Services, Centers Disease Control Prevention, Nat. Center Health Statist., Hyattsville, MD, USA, Tech. Rep. 236, 2016.

[2] A. Bahga and V. K. Madisetti, "A cloud-based approach for interoperable electronic health records (EHRs)," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 5, pp. 894–906, Sep. 2013.
[3] G. Fernández-Cardeñosa, I. de la Torre-Díez, M. López-Coronado, and J. J. Rodrigues, "Analysis of cloud-based solutions on EHRs systems in different scenarios," *J. Med. Syst.*, vol. 36, no. 6, pp. 3777–3782, 2012.
[4] G. Zangara, P. P. Corso, F. Cangemi, F. Millonzi, F. Collova, and A. Scarlatella, "A cloud based architecture to support electronic health record," *Stud. Health Technol. Inform.*, vol. 207, pp. 380–389, Jul. 2014.
[5] C. Moore, M. O'Neill, E. O'Sullivan, Y. Doröz, and B. Sunar, "Practical homomorphic encryption: A survey," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2792–2795.
[6] *Official Google Blog: An Update on Google Health and Google Powermeter*. Accessed: Jun. 11, 2019. [Online]. Available: https://googleblog.blogspot.com/2011/06/update-on-google-health-and-google.html
[7] *Healthcare Data Breach Statistics*. Accessed: Jun. 11, 2019. [Online]. Available: https://www.hipaajournal.com/healthcare-data-breach-statistics/
[8] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. New York, NY, USA: Penguin Random House, 2016.
[9] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *J. Cryptol.*, vol. 3, pp. 99–111, Jan. 1991. doi: 10.1007/3-540-38424-3_32.
[10] D. Bayer, S. Haber, and W. S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," in *Sequences II*. New York, NY, USA: Springer, 1993, pp. 329–334.
[11] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[12] *Bitcoin Mining Consumes More Electricity a Year than Ireland*. Accessed: Jun. 11, 2019. [Online]. Available: https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland
[13] *Bitcoin Energy Consumption Index—Digiconomist*. Accessed: Jun. 11, 2019. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption
[14] M. Scherer, "Performance and scalability of blockchain networks and smart contracts," Ph.D. dissertation, Umeå Univ., Umeå, Sweden, 2017.
[15] D. M. Rind, I. S. Kohane, P. Szolovits, C. Safran, H. C. Chueh, and G. O. Barnett, "Maintaining the confidentiality of medical records shared over the Internet and the world wide Web," *Ann. Internal Med.*, vol. 127, no. 2, pp. 138–141, 1997.
[16] R. Schoenberg and C. Safran, "Internet based repository of medical records that retains patient confidentiality," *Bmj*, vol. 321, no. 7270, pp. 1199–1203, 2000.
[17] F. Uckert, M. Görz, M. Ataian, and H.-U. Prokosch, "Akteonline—An electronic healthcare record as a medium for information and communication," *Stud. Health Technol. Inform.*, vol. 90, pp. 293–297, Jan. 2002.
[18] R. W. Grant, J. S. Wald, E. G. Poon, J. L. Schnipper, T. K. Gandhi, L. A. Volk, and B. Middleton, "Design and implementation of a Web-based patient portal linked to an ambulatory care electronic health record: Patient gateway for diabetes collaborative care," *Diabetes Technol. Therapeutics*, vol. 8, no. 5, pp. 576–586, 2006.
[19] D. Gritzalis and C. Lambrinoudakis, "A security architecture for interconnecting health information systems," *Int. J. Med. Inform.*, vol. 73, no. 3, pp. 305–309, 2004.
[20] S. Bonacina, S. Marceglia, M. Bertoldi, and F. Pinciroli, "A Web-based system for family health record," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 3652–3656.
[21] L. Ibraimi, M. Asim, and M. Petković, "Secure management of personal health records by applying attribute-based encryption," in *Proc. 6th Int. Workshop Wearable, Micro, Nano Technol. Pers. Health*, Jun. 2009, pp. 71–74.
[22] M. Saravanan, R. Shubha, A. M. Marks, and V. Iyer, "SMEAD: A secured mobile enabled assisting device for diabetics monitoring," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017, pp. 1–6.
[23] 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)*, Oct. 2017, pp. 1–5.
[24] V. Patel, "A framework for secure and decentralized sharing of medical imaging data via blockchain consensus," *Health Inform. J.*, vol. 25, no. 4, pp. 1398–1411, 2018.
[25] A. Juneja and M. Marefat, "Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inform. (BHI)*, Mar. 2018, pp. 393–397.
[26] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *J. Med. Syst.*, vol. 42, no. 7, p. 130, 2018.
[27] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 152, 2018.
[28] X. Zhang and S. Poslad, "Blockchain support for flexible queries with granular access control to electronic medical records (EMR)," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[29] S. Badr, I. Gomaa, and E. Abd-Elrahman, "Multi-tier blockchain framework for IoT-EHRs systems," *Procedia Comput. Sci.*, vol. 141, pp. 159–166, Jan. 2018.
[30] 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.
[31] J. Zhang, N. Xue, and X. Huang, "A secure system for pervasive social network-based healthcare," *IEEE Access*, vol. 4, pp. 9239–9250, 2016.
[32] J. Brogan, I. Baskaran, and N. Ramachandran, "Authenticating health activity data using distributed ledger technologies," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 257–266, Jan. 2018.
[33] A. F. Hussein, N. Arunkumar, G. Ramírez-González, E. Abdulhay, J. M. R. Tavares, and V. H. C. de Albuquerque, "A medical records managing and securing blockchain based system supported by a genetic algorithm and discrete wavelet transform," *Cogn. Syst. Res.*, vol. 52, pp. 1–11, Dec. 2018.
[34] 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 (OBD)*, Aug. 2016, pp. 25–30.
[35] 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.
[36] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, "Blockchain-based data preservation system for medical data," *J. Med. Syst.*, vol. 42, no. 8, p. 141, Aug. 2018.
[37] 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.
[38] T. Dey, S. Jaiswal, S. Sunderkrishnan, and N. Katre, "HealthSense: A medical use case of Internet of Things and blockchain," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2017, pp. 486–491.
[39] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, p. 218, 2016.
[40] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32700–32726, 2018.
[41] *What is Practical Byzantine Fault Tolerance (pBFT)*. Accessed: Jun. 12, 2019. [Online]. Available: https://crushcrypto.com/what-is-practical-byzantine-fault-tolerance/
[42] Financial Conduct Authority, "Discussion paper on distributed ledger technology," Financial Conduct Authority, London, U.K., Discuss. Paper DP17/3, 2017.
[43] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[44] *Remote Procedure Call*. Accessed: Jun. 12, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Remote_procedure_call
[45] *Web API*. Accessed: Jun. 12, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Web_API
[46] *Representational State Transfer*. Accessed: Jun. 12, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer
[47] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Proc. Conf. Theory Appl. Cryptograph. Techn. (CRYPTO)*, London, U.K.: Springer-Verlag, 1988, pp. 369–378.
[48] M. Swan, *Blockchain: Blueprint for a New Economy*. Newton, MA, USA: O'Reilly Media, 2015.
[49] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM)*, Jun. 2016, pp. 1–6.
[50] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[51] *Pseudonymity*. Accessed: Jan. 18, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pseudonymity

[52] A. P. Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," *Math. Found. Comput.*, vol. 1, no. 2, pp. 121–147, May 2018.
[53] A. Back. (2002). *Hashcash—A Denial of Service Counter-Measure*. Accessed: Sep. 3, 2016. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf
[54] *Bitcoin Mining Consumes More Electricity a Year Than Ireland|Technology| the Guardian*. Accessed: Dec. 6, 2019. [Online]. Available: https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland
[55] C. Stoll, L. Klaaßen, and U. Gallersdörfer, "The carbon footprint of bitcoin," *Joule*, vol. 3, no. 7, pp. 1647–1661, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2542435119302557
[56] *HIMSS—Healthcare Information and Management Systems Society*. Accessed: Dec. 6, 2019. [Online]. Available: https://www.himss.org/
[57] H. Howard, M. Schwarzkopf, A. Madhavapeddy, and J. Crowcroft, "Raft refloated: Do we have consensus?" *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 12–21, 2015.
[58] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. Newton, MA, USA: O'Reilly Media, 2014.
[59] *Confirmation—Bitcoin Wiki*. Accessed: Jun. 12, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Confirmation
[60] A. Kousaridas, S. Falangitis, P. Magdalinos, N. Alonistioti, and M. Dillinger, "SYSTAS: Density-based algorithm for clusters discovery in wireless networks," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Aug./Sep. 2015, pp. 2126–2131.
[61] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in *Proc. WEIS*, 2013, p. 11.
[62] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99, 1999, pp. 173–186.
[63] E. Brewer, "CAP twelve years later: How the 'rules' have changed," *Computer*, vol. 45, no. 2, pp. 23–29, 2012.
[64] D. Terry, "Replicated data consistency explained through baseball," *Commun. ACM*, vol. 56, no. 12, pp. 82–89, 2013.
[65] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2017, pp. 643–673.
[66] A. Girault, G. Gössler, R. Guerraoui, J. Hamza, and D.-A. Seredinschi, "Why you can't beat blockchains: Consistency and high availability in distributed systems," 2017, *arXiv:1710.09209*. [Online]. Available: https://arxiv.org/abs/1710.09209
[67] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proc. 2nd Int. Conf. Internet-Things Design Implement.*, 2017, pp. 173–178.
[68] A. S. Raghuvanshi, S. Tiwari, R. Tripathi, and N. Kishor, "Optimal number of clusters in wireless sensor networks: An FCM approach," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, Sep. 2010, pp. 817–823.

LEILA ISMAIL completed higher studies (DEA) in distributed systems at the Joseph Fourier University (Grenoble I) and ENSIMAG Engineering University, France. She received the Ph.D. degree (Hons.) in computer engineering/distributed systems from the National Polytechnic Institute of Grenoble, France, in September 2000.

She has a vast industrial and academic experience at the Sun Microsystems Research and Development Center, worked on the design and implementation of highly available distributed systems, and participated in the deposit of a US patent in the domain. She was a Teacher with Grenoble I, France, and an Assistant Professor with the American University of Beirut. She is currently an Associate Professor with the College of Information Technology (CIT), United Arab Emirates University (UAEU), UAE, where she joined, in 2005. She is the Founder and the Director of the Distributed Computing and Distributed Systems Research Laboratory, CIT. She has been serving as an Adjunct Professor with the Digital Ecosystems and Business Intelligence Institute Curtin University, Australia. Her current research interests include performance analysis in distributed systems, energy efficiency and management, green computing, resource management and scheduling problems in distributed systems with emphasis in clouds, middleware, high performance computing, blockchain, and software security in distributed systems.

Dr. Ismail has International collaborations and publishing her research results in prestigious journals and international conferences. She received the IBM Shared University Research (SURA) and the IBM Faculty Awards, very competitive world-wide, and received funding for major projects as a PI/Co-PI and the funded project by UAE/NRF was top ranked by external anonymous reviewers. She served as an Associate Editor of the *International Journal of Parallel, Emergent and Distributed Systems* for several years, served as the Chair, Co-Chair, and Track Chair for many IEEE international conferences, including being a General Chair for IEEE DEST 2009, and a General Chair, Technical Program Chair, and Organizing Committee Chair for the 11th International Conference on Innovations in Information Technology 2015 (IIT'15) for which she got the support of the IEEE Computer Society (HQs) Technical Sponsorship. She is the Editor of *Information Innovation Technology in Smart Cities* (Nature Springer, 2018).

HUNED MATERWALA received the bachelor's degree in instrumentation and control engineering from Gujarat Technological University, India, in 2013, and the master's degree in technology in control systems from the Department of Electrical and Electronics Engineering, Amity University, Uttar Pradesh, India, in 2016. He is currently a Research Assistant with the Distributed Computing and Distributed Systems Research Laboratory, College of Information Technology, UAE University, UAE.

SHERALI ZEADALLY received the bachelor's degree in computer science from the University of Cambridge, England, and the Ph.D. degree in computer science from the University of Buckingham, England. He is currently an Associate Professor with the College of Communication and Information, University of Kentucky. His research interests include cybersecurity, privacy, the Internet of Things, computer networks, and energy-efficient networking. He is a fellow of the British Computer Society and the Institution of Engineering Technology, England.