Received May 25, 2020, accepted June 9, 2020, date of publication June 19, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003575

EdgeMediChain: A Hybrid Edge Blockchain-Based Framework for Health Data Exchange

RAIFA AKKAOUI, XIAOJUN HEI, (Member, IEEE), AND WENQING CHENG, (Member, IEEE)
School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding authors: Raifa Akkaoui (raifa_akkaoui@hust.edu.cn) and Xiaojun Hei (heixj@hust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972172.

ABSTRACT Recently, researchers around the world in medical institutions and pharmaceutical companies are demanding a wider access to healthcare data for secondary use in order to provide enhanced and personalized medical services. For this purpose, healthcare information exchange between health authorities can be leveraged as a fundamental concept to meet these demands and enable the discovery of new insights and cures. However, health data are highly sensitive and private information that requires strong authentication and authorization procedures to manage the access to them. In this regard, the cloud paradigm has been used in these e-healthcare solutions, but they remain inefficient due to their inability to adapt to the expanding volume of data generated from body sensors and their vulnerability against cyberattacks. Hence, collaborative and distributed data governance supported by edge computing and blockchain promises enormous potentials in improving the performance and security of the whole system. In this paper, we present a secure and efficient data management framework, named *"EdgeMediChain"*, for sharing health data. The proposed architecture leverages both edge computing and blockchain to facilitate and provide the necessary requirements for a healthcare ecosystem in terms of scalability, security, as well as privacy. The Ethereum-based testbed evaluations show the effectiveness of *EdgeMediChain* in terms of execution time with a reduction of nearly 84.75% for 2000 concurrent transactions, higher throughput compared to a traditional blockchain, and scalable ledger storage with a linear growth rate.

INDEX TERMS Blockchain, data sharing, edge computing, electronic medical records, healthcare, Internet of Things, privacy, security, smart contracts.

I. INTRODUCTION

The large aging population worldwide and the drastically increasing number of people suffering from long-lasting diseases, like diabetes, have been and still remain a major struggle for health institutions throughout the globe. According to a report published by the world health organization, the population of patients with diabetes escalated from 108 million to over 422 million in 2014. Furthermore, the disease has been directly linked to the death of nearly 1.6 million people in 2016 [1]. In an attempt to alleviate the burden of chronic diseases, several technological solutions have been proposed and deployed to improve the overall delivery of healthcare services, healthcare information exchange (HIE) among health authorities has been proven to be a major improvement factor for the healthcare industry [2]. Not only can HIE strengthen the understanding of clinical trials for individual patients, but it also allows for the extension of scientific discoveries by enabling researchers to pool data from multiple trials for further analysis, which might lead to the discovery of new insights and cures beyond those deducible from any individual study [3].

The electronic medical records (EMRs) shared by healthcare institutions offer great insights for the improvement of the healthcare industry. Nevertheless, it is also of great importance to consider the tremendous amount of personal health data (PHD) generated from wearable personal health devices. With the new emerging eras of Internet of Things (IoT) and big data [4]–[6], many healthcare applications have been developed using body sensors with the main purpose of monitoring biomedical signals of individuals, and they have already generated a great deal of information regarding patients' vital signs on a daily basis. For instance, doctors can potentially use these data for precision medicine [7], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Chi Chen.

More precisely, physicians can take a variety of parameters into consideration, such as the environment, lifestyle, diet, and daily activities, while providing a diagnosis to their patients. Furthermore, pharmaceutical companies can also take advantage of these data to study the impact of certain drugs on the overall recovery and well-being of patients as well as the side effects which might come along with the treatments.

However, it is crucial to acknowledge that storing and sharing such huge amount of data is challenging on several fronts. In this context, the cloud computing paradigm has been used for more than a decade as the de facto for processing and sharing this information. These cloud service providers (CSPs) offer different solutions seen as being reliable and efficient for data storage and processing. Yet, they are currently struggling to meet these requirements. In point of fact, the storage market based on the cloud is currently suffering from a lack of fair competition and unwillingness of sharing data, as it is mainly dominated by few big tech companies such as Google, Amazon, Microsoft, etc [9]. Furthermore, using cloud-based solutions for the deployment of health applications is seen as unfit for they require real-time processing and delays can be a critical issue as it might result in failures or a misdiagnosis [10]. To tackle this, edge computing has emerged as an alternative paradigm for using cloud computing. Precisely, by bringing computation and storage capacities closer to the end-user layer and data sources, the paradigm helps mitigating delays and latency [11], [12]. Hence, edge computing can be seen as a remedy to solve some of the CSPs limitations and enhance the deployment of efficient and effective technological healthcare solutions. However, healthcare data are highly sensitive and contain critical information related to the patient's private life [13]–[15]. Therefore, to avoid any malicious exposure strong authentication and access control (AC) policies need to be guaranteed, in order to avoid unauthorized access. In this perspective, several blockchain-based systems have been proposed [16]–[21], by leveraging the powerful features of the technology such as decentralization, consensus protocols, and immutability to ensure the protection of these private data. For instance, in [22] a Hyperledger-based EMRs sharing framework was proposed which utilizes chaincode to enforce access policies. Meanwhile, the authors in [23] integrated the blockchain technology with deep learning techniques to manage the process of sharing EMRs between multiple healthcare entities. However, all these solutions remain bound to the scalability issues of the growing ledger size and the transaction per second rate, specifically, while merging a blockchain-based system with an IoT-based ecosystem in which IoT devices are expected to generate tremendous volumes of transactions.

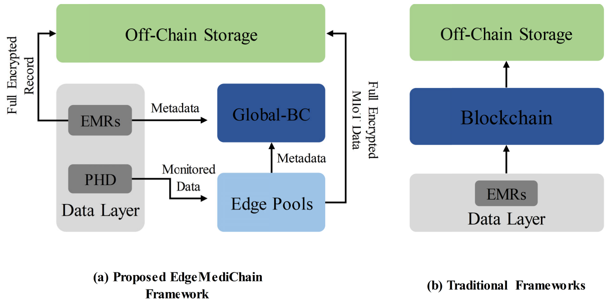
A. MOTIVATION

Despite the abounding advantages of blockchain in terms of security, it still faces some challenges limiting its extensive usage, as in some cases the shift to a distributed network may not make that much sense and even if such changeover is profoundly beneficial, the requirements of certain applications might go against what blockchain can offer and the network might not be able to fulfill them alone. It is without doubt that blockchain and smart contracts bring an abundance of assets to the equation, however, they also come with some disadvantages. Precisely, blockchain networks suffer from a trilemma in which a blockchain-based system can only have at most two of the following features: decentralization, scalability, and security [24], [25]. In point of fact, the scalability issue, precisely in terms of low throughput, high latency, resource draining, and ledger height, lower the practicality of any blockchain-based system on a large-scale. Actually, as the number of processed transactions builds up the storage space required for the immutable ledger increases drastically. For instance, the size of the bitcoin blockchain has been experiencing consistent high levels of growth since its creation, reaching relatively 210 GB in size as of April 2019. The Ethereum blockchain is not immune to this issue, however, the amount of growth is moderate as Ethereum only saves the state instead of the whole blockchain history at each full node.

The aforementioned limitations make us raise the following question: **Can we divide the process of mining transactions between narrower parallel groups of nodes closer to the source of data in order to increase the blockchain's total throughput, while still ensuring the security of the overall system?** Actually the answer to this might reside in the emerging technology, namely, edge computing. The concept of the paradigm is based on the idea of separating data to store it and process it locally near the data source across different distributed locations. However, there is no mechanism that can guarantee the integrity of the data stored separately within several edge nodes (ENs) which can be compromised due to risks of losing data, having incorrect storage or malicious adversaries. In contrast, the blockchain ledger is shared among all nodes of the network, in which every block has to be mined and voted before being added to the chain to never be erased or altered, which ensures the integrity of the data. Therefore, it is clear to see the benefit of combining both technologies to solve the limitations of one another. Specifically, the possibility of integration is rooted in the fact that both networks are based on decentralized infrastructures, meanwhile, the need of integration comes from the diverse advantages of blockchain and edge computing as well as their joint interdependent features. Hence, by combining both technologies in our system we aim to provide a secure framework to fulfill the requirements of a healthcare data sharing ecosystem, while also taking into account the network storage and computational resources, which meet the essence of blockchain and the overall capacity of edge computing.

B. CONTRIBUTIONS

In this work, we propose the design of a four-layered framework, namely *EdgeMediChain* illustrated in Fig. 1, which aims to facilitate the process of sharing health data (i.e., EMRs and PHD), by combining both the attractive features



(a) Proposed EdgeMediChain Framework

(b) Traditional Frameworks

FIGURE 1. Proposed EdgeMediChain vs traditional frameworks.

of edge computing as well as blockchain in order to meet the requirements discussed previously. Differently than the already existing blockchain-based frameworks in the literature [26]–[28], *EdgeMediChain* leverages the edge computing paradigm to ensure availability and parallelized high performance, precisely, in terms of handling the astonishing amount of monitored data generated from the IoT devices and body sensors, meanwhile, the blockchain technology was adopted to provide the required privacy and security in the process of sharing health data. We extend our work in [29], by introducing a new source of data (i.e., IoT body sensors) and focusing on the main idea of integrating a set of local consortium blockchains, namely local edge-mining pools, to manage the IoT devices' accounts, authentication, and data storage (i.e., the behavior of the IoT devices). Smart contracts are also utilized to ensure an automated regulation vis-à-vis AC rules and policies governing the access to the shared health data in a decentralized and non-deniable way. The proposed solution ensures that all activities and transactions to access patients' data are chained to the blockchain ledger for secure data logging and auditing. As a short summary, the key contributions of our proposed work are as follows:

1) We present the proposed four-layered architectural design of our hybrid edge blockchain-based distributed health data-sharing framework for transparent and secure HIE as we discuss the threat model of the system, the details of its components and their interactions.
2) We develop a set of smart contracts for autonomous decision-making, precisely in terms of the authentication of patients and their IoT devices to ensure they are certified and patched in a secure and correct manner as well as in terms of restricted AC policies, we also evaluate the proposed mechanisms including their overall functionalities and workflow with a case study.
3) We implement a prototype based on the Ethereum blockchain platform to validate and evaluate the feasibility as well as the performance of the proposed *EdgeMediChain* architecture that we compare against a traditional approach, we also evaluate the performance of our proposed architecture under different configurations.
4) We then analyze the security performance of our proposed authentication/authorization schemes in terms of confidentiality, integrity, transparency, and privacy.

C. ORGANIZATION

The remainder of this paper is organized as follows: Section II briefly introduces the blockchain preliminaries. In Section III, we present the architectural design of our proposed edge-blockchain data-sharing framework and discuss its layers and components. In Section IV, the authentication and authorization smart contracts are presented, including their overall functionalities and detailed workflow. The performance results as well as the security aspects of *EdgeMediChain* are analyzed in Section V. We then present a taxonomy of the state-of-the-art in terms of blockchain-based solutions dedicated for health data sharing and solutions addressing the scalability issue while combining blockchain with an IoT ecosystem in Section VI. Finally, Section VII concludes this paper with some future work.

II. PRELIMINARIES

One definition we can attribute to blockchain is that it is a scattered and disperse ledger that is used to archive transactions among different trustless entities in an efficient, confirmable, and indestructible way. The network is merely based on a set of peer-to-peer (P2P) nodes jointly complying to a consensus protocol to manage transactions and validate new blocks. As blockchain is based on Merkle trees, this ensures that no entity would be able to alter the data within a block without it altering all chained blocks, which requires controlling the majority of the mining power. It was in 2008 when Satoshi Nakamoto (note that this is the pseudonym used to refer to the person(s) who built Bitcoin) came up with the idea of blockchain for the first time as the building block of the Bitcoin cryptocurrency [30]. The major goals behind the technology are to offer secure, pseudo-anonymous, reliable, transparent, and trustworthy transactions among trustless individuals, without the need of a trusted third party (TTP). The core fundamental base of blockchain is a decentralized ledger shared between all nodes of the network rather than it being controlled by a central authority. Furthermore, consensus protocols are also among the major pillars of blockchain by which the majority of nodes agrees on a definitive conclusion whether to add a block to the chain. Ensuring that each generated transaction is certified, mined, and approved before being added to never be erased to the ledger by chaining it with a hash pointing to the previous block (i.e., hash pointer). During the recent years, bitcoin was able to attract a tremendous amount of developers and researchers which utilized the appealing features of the cryptocurrency technology to solve its excessive computation overhead in terms of variations of proof-of-work (PoW), such as proof-of-authority (PoA), practical Byzantine fault tolerance (PBFT) and many other protocols, or regarding scalability using different hashing algorithms. Apart from Bitcoin, there exists a variety of blockchain frameworks, in this paper, we leverage the Ethereum blockchain due to its attractive assets, including flexibility, completeness, maturity, availability of its development tools, and its Turing complete Ethereum virtual machine (EVM) making it a programmable blockchain utilizing smart contracts written in Solidity [31] stored on-chain and with no size restrictions.

A. ETHEREUM

Ethereum is quite similar to bitcoin in a sense that both can be seen as decentralized permissionless blockchain networks. However, they differ in some major technical aspects which are their purpose and capacities. On the one hand, Bitcoin is merely a P2P cryptocurrency system that allows electronic transactions between two parties. On the other hand, Ethereum, as indicated before, is a programmable blockchain that focuses on the idea of running codes in a decentralized manner, which allows for the development of decentralized applications with no single point of failure, also known as DApps [32]. Ethereum uses Ether (ETH) as its own cryptocurrency, in which miners are solving puzzles to earn ETH rather than bitcoin. As of June 2020, Ethereum has a market capitalization of 27.374 billion USD and a market value per ETH of approximately 245.95 USD [33]. The platform is also based on a second form of token, namely *gas*, which is needed upon each smart contract execution to incentivize miners to mine and chain new transactions (the notions of both gas and smart contracts will be further detailed in Subsection II-B and II-C). Furthermore, Ethereum offers two categories of accounts: an *externally owned account* (EOA) and a *contract account* (CA) which are identified by 20-bytes addresses. Any change regarding the state of the Ethereum blockchain is initiated by an EOA as illustrated in Fig. 2. This means that the only way to interact with the code of any given CA is by initiating a transaction from an EOA that includes the input parameters needed for the CA code execution. Moreover, Ethereum offers a specific type of operations, which enable querying information from the blockchain without any fees referred to as *Calls* which are used to query the IoT devices' data in our proposed framework.

B. GAS AND PAYMENT

The notion of *gas* is particularly a crucial concept within the Ethereum blockchain, it specifies that **for each computation triggered by the execution of a transaction within the network some fees have to be paid and those fees are what is called gas**. In other words, *gas* is the unit utilized to evaluate the required fees upon the execution of any particular computation. Precisely, *gasPrice* is the quantity of ETH a user is deliberately choosing to allocate to each unit of gas and is measured in *"gwei"*, where 10^{18} Wei represents 1 ETH and 1 gwei is equal to 10^9 Wei. With every transaction, a sender sets a *gasLimit*, a *gasPrice*, and the product of the two represents the maximum amount of Wei a sender is willing to spend so that his transaction is executed. For instance, if a sender sets the *gasLimit* to $50,000$ and the *gasPrice* at $20\ gwei$, this implies that the sender is willing to spend at most $50,000 \times 20\ gwei = 10^{15} Wei$, where $1\ Wei = 0.001\ ETH$ to execute that transaction [32].

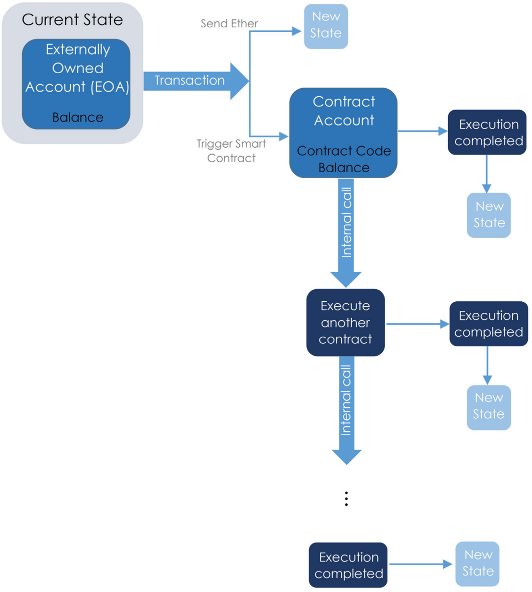


FIGURE 2. Ethereum transaction workflow diagram.

The main reason behind gas is that Ethereum is a Turing complete machine, which allows for loops that make the network susceptible to cyberattacks, as it's hard to conclude whether a program will run infinitely. Hence, if fees didn't exist, an attacker could easily saturate the network by trying to execute a transaction containing an infinite loop, without any backlash. Therefore, gas defends the network from malicious nodes trying to launch distributed denial of service (DDoS) attacks.

C. SMART CONTRACTS

Ethereum allows for building trustless systems, as different entities whom do not trust each other necessarily can send transactions within the P2P network, which enables faster reconciliation between the different transacting participants. Furthermore, the usage of cryptography, being the main feature of blockchain, achieves indubitableness behind all the transactions in the network. Precisely, this behavior can be attained using smart contracts, which are self-executing scripts residing on-chain, allowing for proper, disperse, and densely automated workflows. Nick Szabo introduced the concept of a smart contract and defined it as being a digital protocol that allows for the automated execution of a contract's terms [34]. Actually, in order to lower the need for a TTP, as well as the occurrence of mischievous or unintentional exceptions and self-enforce contractual clauses, Szabo proposed translating these clauses into code which will be then embedded into hardware or software properties. In the case of blockchain, smart contracts are actually scripts stored on the EVM with a unique address (i.e., CA) allowing users (i.e., EOA) to interact with them by initiating a transaction with the smart contract address, which triggers an independent and automatic execution of the code on every node of the blockchain network based on the function called within the transaction.

III. SYSTEM DESIGN

In this section, *EdgeMediChain*'s hierarchical architecture is presented, as we discuss the threat model and the assumptions upon which the system is built, we also highlight the different layers of the proposed framework, their respective components, and how they interact between each other.

A. THREAT MODEL AND ASSUMPTIONS

In the following subsection, we discuss the proposed framework threat levels in terms of authentication, access rights, integrity, and anonymity, our contributions as well as the assumptions upon which the system is built.

1) AUTHENTICATION AND OWNERSHIP

Currently, medical IoT (MIoT) devices can take a wide variety of forms, such as, wearable, implantable, or injectable. However, patients' lives can be at a huge risk with the rise of counterfeited MIoT devices. As a tremendous number of wearable IoT devices is expected to join the network, it is crucial to ensure that all of them are authenticated and to guarantee a trusted proof-of-ownership and firmware verification in a decentralized and secure manner. To tackle this, every device in our proposed system has a public and private key pair generated using the elliptic curve digital signature algorithm (ECDSA) as well as an Ethereum identifier generated from the *keccak-256* hash of the last 20-bytes of the device's owner public key. Furthermore, the overall management of the MIoT devices is governed in a decentralized manner without a single point of failure using the proposed *MIoT-Edge-Manager* smart contract, which we will further detail in Subsection IV-A.

2) ACCESS RIGHTS

Various degrees of access to data should be granted to different authorized entities. Hence, the need to allow access to the parts of the medical records which are only relevant to the practitioners or even the patient under the "need-to-know" principle in conformity with their clinical obligations or functions. In this paper, we propose an authorization smart contract (presented in Subsection IV-B) which is compliant with the health insurance portability and accountability act (HIPAA) [35] and guarantees that every participant is allowed to access the shared data based on the *minimum necessary* rule, as data requestors need to initiate a challenge-response procedure to prove their access rights according to their roles defined in the smart contract. In addition, the unalterable feature of the blockchain shared ledger will guarantee a strict and deep monitoring of all the access requests to the shared data in *a tamper-proof manner* by continuously triggering the execution of the smart contract.

3) CREDIBILITY

In an information driven world, it is crucial to ensure that patients have full ownership over their health data, however, we still need to verify the integrity of the shared records uploaded if we want to ensure that the system will serve assiduously its intended goals. Hence, a *multisignature* smart contract is implemented to provide patients the ownership over their shared EMRs keeping the system patient-centric, but requires the records verification. Precisely, during the registration phase the proposed smart contract requires the validation of the shared record *only by* the doctor who issued it, this is not a violation of the HIPAA in anyway, as it is actually doctors who are generating those medical records in the first place, therefore, they also have the ownership right to them. However, if a patient wants to see a different doctor, a delegated approval is needed before granting access to the EMR.

4) ANONYMITY

Recently, people are becoming more and more concerned about the privacy of their data, particularly health data, which are characterized by a high level of sensitivity for they deal with personal details that most individuals aren't willing to share for a countless number of reasons unless incentivized to do so. Therefore, as patients might be reluctant to whether they should share their EMRs for privacy concerns, a possible solution to this problem is *data anonymization*, which once applied it guarantees the privacy of the data and the identity of patients to whom the data belong. In point of fact, according to a recent study, 64% of patients who were part of the questionnaire reported that they are comfortable with the confidential sharing of their health data as long as personally identifiable information (PII) is omitted [36]. In this regard, PII is supposed to be left out (e.g., names, street addresses, neighborhood, dates of birth, phone numbers, email addresses, social security IDs, EMRs numbers, bio-metric IDs, photos or any identifiable images, and genetic data) and only non-PII such as age, gender, non-specific geographical locations (e.g., city, province, country), and summary/partial health data are to be shared.

Finally, the deployed blockchain layers are assumed to be consortium in which identities of nodes that participate as miners on the network are assumed to be verified off-chain. For instance, a local health certificate authority (HCA) might be leveraged to enable a process allowing the registration of miners which will then associate them with their respective organizations. Patients also need to register off-chain through a secure channel in order to get their personal accounts and respective accounts for each MIoT device they own as well as a secret shared key (S_k) that will be then used for the symmetric encryption of their health data (both EMRs and PHD). We also assume that it is impossible to crack standard cryptographic primitives (e.g., finding hash collisions, forging digital signatures, etc.). Moreover, as our proposed system is mainly built on blockchain any user is not allowed

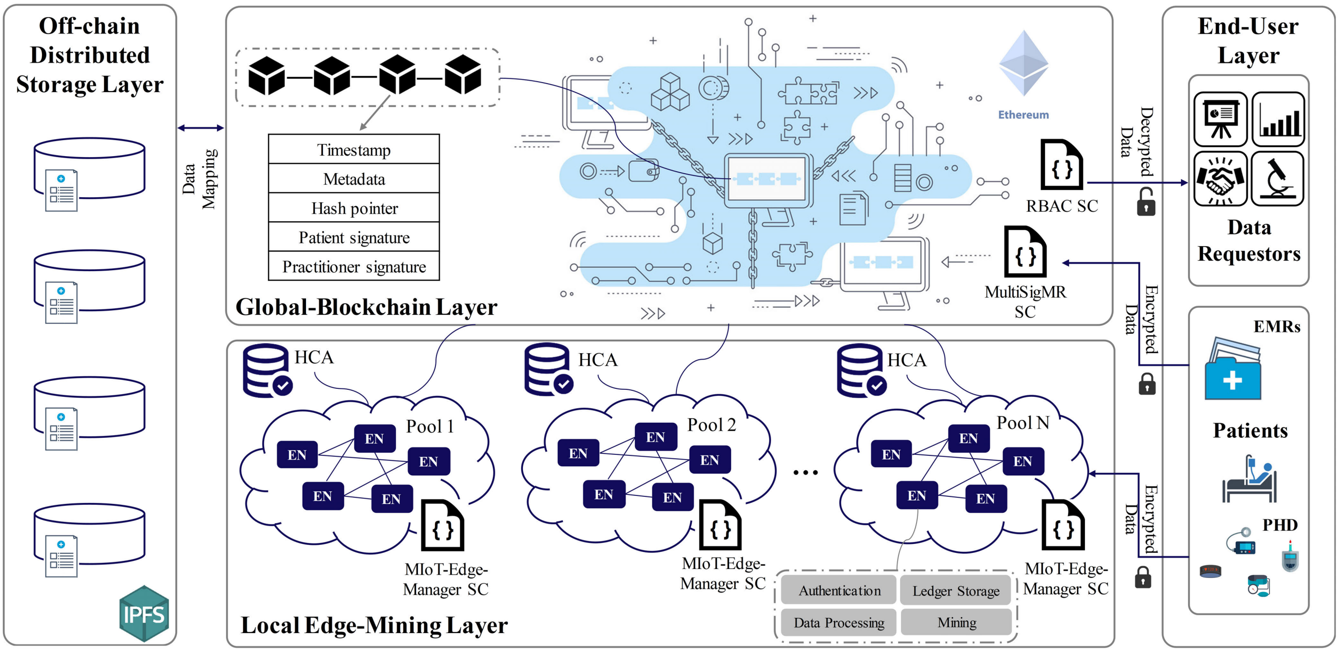


FIGURE 3. EdgeMediChain four-layered architecture.

to control the majority of the mining power to prevent the double-spending attack and the 51% attack [37].

B. SYSTEM ARCHITECTURE

As we have discussed the threat model and assumptions of the proposed framework. In this subsection, we introduce the hierarchical architecture of *EdgeMediChain*, which is composed of four main layers as illustrated in Fig. 3. The different layers are independent and decentralized in terms of computation and storage management, yet jointly interconnected and scheduled to achieve the overall performance of the system with a higher scalability, reliability, and traceability. Precisely, the lower layer has a wide set of health data generators and consumers. Then we have two middle layers, a local edge-mining layer governed by health authorities that deploy various ENs to process the PHD within a zone and are in charge of determining the capacity as well as the number of nodes within each local mining pool, and a global-blockchain layer that allows endorsement for data requestors to access healthcare data within the permissions and rules defined in the smart contract deployed. Last, the off-chain distributed storage layer is responsible for maintaining the full encrypted data.

The whole architecture is based on a parallelization scheme which aims to increase computational capabilities with regards to the MIoT devices and at a lower transaction cost. Transactions generated from body sensors are offloaded across multiple local mining pools (or shards), processed in a parallel manner with low latency, and accessed via smart contracts deployed on the global-blockchain network which define a set of permissions and AC policies, while maintaining data utility and avoiding privacy disclosure simultaneously. Therefore, latency is minimized by enhancing the real-time processing, but also scalable data analysis, strong security policies, and convenient data sharing are guaranteed, meeting the requirements of a HIE system. Furthermore, some health coins can be leveraged by external data requestors to incentivize patients (i.e., data owners) eventually building a sustainable health data market, however, we should note that this is beyond the scope of this paper. In what follows, we present the details of the components within each layer of the proposed hierarchical architecture.

1) END-USER LAYER

Participants in our proposed framework are classified under two categories (i.e., data generators and data requestors) who can submit, share, and access health data. Furthermore, the shared data are separated into EMRs and PHD due to the discrete requirements to share them. Precisely, EMRs (e.g., diagnostic reports, laboratory results, medical examination reports, X-Ray scans, etc.) are privacy sensitive data but with moderate latency requirements. Furthermore, the credibility of these data is strongly needed, as an incorrect medical report can be life-threatening to the patient. Therefore, if the EMR published upon the visit of a patient to a hospital is signed using both parties signatures, none of them can refute the treatment. Meanwhile, quantity as well as privacy are the major concerns of sharing PHD, as the volume of health data produced by each patient using body sensors is astonishing. Hence, it is clear to see that the corresponding requirements are undoubtedly different. For this purpose, we propose to manage PHD and EMRs using a local and a global-blockchain respectively.

2) LOCAL EDGE-MINING LAYER

Entities deploying over blockchain networks need to process and store a large quantity of information that is of no interest to them. Upon the first launch of a new node downloading, verifying, and storing the entire history of all transactions is needed, although most of these transactions are not relevant to the node. Hence, to avoid this cumbersome, we deploy a local edge-mining pools distributed structure. Each local edge-mining pool is composed of a set of mining nodes (i.e., ENs) that manage the patients' PHD within a geographical location, each pool is responsible for registering patients and their devices on-chain, receiving data generated from those body sensor devices and it is also in charge of the authentication procedure to manipulate the data. We should note that the ENs are managed by sealer(s) using the PoA consensus mechanism which does not require any mining resources as it is based on authorized sealers responsible for the validation of new blocks rather than solving highly computational mathematical puzzles.

Meanwhile, to record data generated from the MIoT devices globally the ENs are also part of the global-blockchain as lightweight nodes, however, they don't continuously send transactions to mitigate ETH overuse. Hence, data needs to be analyzed and evaluated locally to adjust the recording time reasonably. If we take the case of a patient with diabetes using a continuous glucose monitoring MIoT device, the sensor would measure the blood sugar levels of the patient, which according to the American diabetes association should be around 80 to 130 mg/dL before eating a meal and less than 180 mg/dL about 1 to 2 hours after eating a meal, the sensor would test the glucose every few minutes and a transmitter wirelessly will send the information to the personal node (PN) of the patient to generate a transaction that would be then sent to the local edge-mining pool containing the encrypted data. The ENs could then process the data off-chain using machine learning techniques and might only send for instance abnormal levels to the global-blockchain [38].

3) GLOBAL-BLOCKCHAIN LAYER

This component operates as a blockchain database, by storing the hashes of the EMRs generated using *keccak-256* as well as URL hash pointers. This ensures that the detailed EMRs are not publicly accessible hence preserving the privacy of patients. In addition, EMRs are heavy files of several megabytes and storing them on-chain requires a high throughput and storage resources. Therefore, only the hash value, which is of a fixed size around several kilobytes, is stored on-chain. Meanwhile, integrity is preserved, as data requestors can verify the credibility of the shared data to authenticate the EMRs. Ensuring the non-repudiation for both the hospital and the patient. Furthermore, this layer consists of the processing and consensus nodes (i.e., managers) responsible for the execution of the authorization smart contract -managing the access to the shared data- as well as blocks mining based on the PoW consensus mechanism. Furthermore, each node is supposed to keep a copy of the shared ledger monitoring all the data request transactions.

4) OFF-CHAIN DISTRIBUTED STORAGE LAYER

Blockchain cannot guarantee privacy and transparency simultaneously. Precisely, storing raw data on-chain will spark great privacy concerns as well as scalability issues. Hence, to ensure privacy and authenticability simultaneously, we propose to leverage both off-chain storage and on-chain verification with the off-chain storage being mainly responsible for storing the complete set of records. For this cause, the interplanetary file system (IPFS) [39] protocol can be leveraged, which is a P2P distributed protocol that aims to attach all computing devices with the same system of files with no single point of failure. The protocol leverages concepts from preceding P2P systems similar to BitTorrent, Git, Self-certified File Systems, and distributed hash tables and group them to form a unique homogeneous framework dedicated to distributing heavy data. Furthermore, it is interoperable with Ethereum smart contracts, hence, it can add reliable and low-cost storage capacity to our proposed blockchain-edge ecosystem.

IV. PROPOSED EDGE-BASED MIoT AUTHENTICATION AND ROLE-BASED AUTHORIZATION SCHEMES

In this section, we provide the details of both the authentication and authorization smart contracts proposed which are responsible for managing the MIoT devices activities and the access control to the shared health data respectively.

A. MIoT DEVICES AUTHENTICATION MECHANISM

With the new emerging area of intelligent healthcare, authenticity and reliability of the MIoT devices utilized by patients are of tremendous concerns. Nowadays, these devices come in different shapes and sizes scaling from wearables, implantables, or injectables. This advancement has made the life of patients at a higher risk in case they are victims of counterfeited MIoT devices. Hence, it is crucial to have the ability to verify those devices in a reliable, trusted, and auditable way with no centralized management. Furthermore, data manipulation and hijacking of MIoT devices in the network are of no less importance [40]. Hence, security policies and restrictions should be developed and maintained. However, the currently deployed IoT systems rely heavily on centralized authentication techniques, both in design and deployment and are based on TTPs (e.g., the open authorization protocol). But these methods represent a single point of failure and remain ineffective in terms of cost, security, and privacy.

To mitigate the limitations of the already existing centralized techniques for authentication, a rather fully decentralized authentication scheme using ENs and blockchain is proposed in this paper. The proposed method eases the management of the MIoT devices at scale, by taking advantage of the PNs

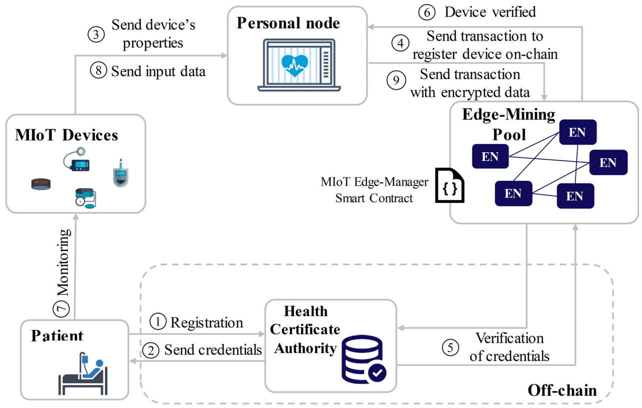

③ Send device's properties
⑧ Send input data
Personal node
⑥ Device verified
④ Send transaction to register device on-chain
⑨ Send transaction with encrypted data
MIoT Devices
Edge-Mining Pool
EN
EN
EN
EN
EN
MIoT Edge-Manager Smart Contract
{}
⑦ Monitoring
Patient
① Registration
② Send credentials
Health Certificate Authority
⑤ Verification of credentials
Off-chain

FIGURE 4. **Proposed MIoT-Edge-Manager authentication procedure.**

and the ENs deployment while also providing the required security needs for a healthcare ecosystem. The procedure is illustrated in Fig. 4, which depicts the workflow of the proposed authentication mechanism between the different components of the system. The first steps of the proposed scheme are 1 and 2, in which each patient registers with his own MIoT devices via a HCA to get the pairs of public and private keys as well as the Ethereum accounts associated with the patient and his devices. Then comes step 3 and 4, in which the patient retrieves the properties and firmware hash of his devices that he sends in a transaction to the local mining pool via his PN using the *DeviceRegistration()* function defined in the *MIoT-Edge-Manager* smart contract. The ENs will verify the information provided (step 5 and 6) only then after its validation the patient can start sending his monitored health data to the edge-mining pool (step 7, 8, and 9) to be further processed locally.

The pseudocode of the proposed *MIoT-Edge-Manager* smart contract is detailed in Algorithm 1. Each MIoT device is defined by the following variables: an *account* which is an *Ethereum address* that represents the actual EOA of the device retrieved during the registration phase, an *EthIdentifier* that is a unique *bytes32* type generated by taking the last 20-bytes from the *keccak-256* hash function applied to the public key of the owner of the device. In Fig. 4, the *EthIdentifier* is used to verify whether the device belongs to the given owner by checking the *OwnerDevice* mapping defined within the smart contract. Then we have the *MIoTProperties* and *Hashfirmware* a *bytes32* type which represent respectively the properties (e.g., a unique identifier of the device or a serial number provided by the manufacturer) and the hash of the compiled firmware file of the MIoT device. These variables are used to ensure the reliability of the MIoT devices as each edge-mining pool is assumed to have an off-chain association with some devices manufacturers to verify whether the provided properties of the device are authentic and whether the firmware is up-to-date. Finally, there is the *MIoTdata* which is a *string* representing the encrypted data generated from the wearable MIoT device (e.g., sensor data such as blood pressure, temperature, blood glucose level, heart rate, etc.) using the secret key S_k. We also provide a mapping *CountofDevices* between *addresses* and *uint* to keep track of the number of MIoT devices owned by each patient, we introduce the modifiers *onlyby()* and *onlybySealer()*, where the first restricts the execution of some functions only by the actual address of the device, whereas the latter restricts the execution of some functions only by the actual address of the ENs (i.e., authorized sealers). Among the functions defined in our proposed smart contract are *DeviceRegistration()* which is responsible for the registration of the MIoT device on-chain and returns a *uint* called *MIoTId* that represents the index of the array used to store the devices within the local edge-mining pool. Other functions are *ValidateHashFirmware()* and *MIoTdataUpdate()*, the first can be executed only by the verified sealers within the local edge-mining pool, whereas, the latter can only be executed by the account associated with the device to update the stored data, we should note that it allows users to upload data from their MIoT device *only and only if* its firmware hash has been confirmed (i.e., up-to-date) using the defined *ifValidated()* modifier.

Algorithm 1 MIoT-Edge-Manager Smart Contract

```
Require: Initiation of mappings and parameters: OwnerDevice, CountofDevices, Hashfirmware, MIoTProperties, ValidHash;
Ensure: Setting up of modifiers: onlyby(); onlybySealer(); ifValidated();
  function DeviceRegistration(EthIdentifier, MIoTProperties, Hashfirmware)
    if (OwnerDevice[msg.sender] == EthIdentifier) then
      Allocate an array to store the device's variables on-chain;
      return MIoTId;
    else
      revert();
    end if
  end function
  function ValidateHashFirmware(MIoTId, MIoTProperties, Hashfirmware) onlybySealer()
    if (firmware is valide) then
      ValidHash = true;
      emit DeviceValid();
    else
      emit DeviceNotValid();
    end if
  return ValidHash;
  end function
  function MIoTdataUpdate(MIoTId, newMIoTdata) onlyby() ifValidated()
  update data;
  emit MIoTDataUpdated();
  end function
```

B. ROLE-BASED AUTHORIZATION MECHANISM FOR DATA ACCESS

In order to improve health services, access to patients' medical information has to be granted to either physicians, medical students, or others who might be involved for any legitimate reason in the patients' care or by other research departments of healthcare institutions such as laboratories, pharmaceutical, and big tech companies. Hence, the need for an authorization scheme which will ensure granting access to only certain information in a distributed manner with no central authority. Precisely, blockchain is the right choice of technology that offers this approach, due to its decentralized nature, AC polices are distributed across different nodes, which offers a decentralized, transparent, and available ledger for storing and granting access to the shared records in an immutable manner. Furthermore, the dynamicity of any healthcare ecosystem makes a role-based access control (RBAC) scheme an efficient and plausible choice for our system. The mechanism defines an AC model for managing users' access to the resources of a given system, based on the concepts of roles and privileges [41]. The model is built upon four main blocks within each a number of features are provided to the whole mechanism. The first block is composed of five components, in which the definition of access is structured using three sets (S for Subjects, R for Roles, and O for Operations) and two relations (Subject-Role assignment $SA \subset S \times R$ and Role-Operation assignment $OA \subset R \times O$). Where, a subject s can perform a given operation o *if and only if* there is a role r such that $(s, r) \in SA$ and $(r, o) \in OA$.

As mentioned previously the consortium blockchain-based framework is under the governance of HCAs required to determine the needed managers within the system responsible for the deployment and initiation of the smart contracts. Meanwhile, each health organization (e.g., hospitals, private clinics, pharmaceutical companies, research institutions, etc.) consenting to be part of the data-sharing framework would be responsible for the generation of the Ethereum addresses representing all the users and their specific roles they are required to be managing. It should be noted that the creation of these addresses including the associated pair of keys can be achieved using multiple approaches such as online or offline generators. Then, each organization would publish the list of addresses and their specified roles without providing any additional details about the identity of users or secret keys for privacy concern. Furthermore, by publishing these Ethereum addresses they would serve as a tool to verify whether such specific address or role is under the management of a given organization (logically no organization would benefit from sharing unauthentic addresses, hence it is assumed that any published Ethereum address indeed belongs to the specific organization). The next step is for the association of each practitioner with his role by the designated consensus nodes (i.e., managers) using the RBAC smart contract (i.e., *role_Assignment()* function).

Algorithm 2 RBAC Smart Contract

```
Require: Initiation of mappings and parameters: UserMap,
  PractiMap, B_List, Rmap, manager;
Ensure: Setting up of modifiers: onlyBy(); PreConditions();
  onlyIf();
  function addUser(address) PreConditions()
      if (PractiMap[address] == address($0 \times 0$)) then
          update mapping of users;
      else
          already a practitioner;
      end if
  end function
  function addPractitioner(address) PreConditions()
  update mapping of practitioners;
      if (UserMap[address] != address($0 \times 0$)) then
          remove address from users mapping;
      end if
  end function
  function ChangeManager(address) onlyBy()
      if (PractiMap[address] != address($0 \times 0$)) then
          manager = address;
      else
          revert();
      end if
  end function
  function role_Assignment(address, role) PreConditions()
      if (PractiMap[address] != address($0 \times 0$)) then
          call ModifyStateOfRole(role);
      end if
  end function
  function RecordAccess onlyIf()
      if (permi[address] != 0) then
          access granted;
      end if
  end function
```

In addition, according to the level of privilege of a role a practitioner can delegate the access right to another user or add other users to the system. Meanwhile, patients can use the same credentials they retrieved upon their registration with the HCA in the authentication scheme to interact with the RBAC smart contract. The pseudo-code of the proposed smart contract detailed in Algorithm 2 is an implementation of the RBAC model which introduces a mapping *Rmap* between roles and permissions in order to assign for each user a role delegated permission based on his affiliation and in accordance with the clinical needs, mappings between users and their contract accounts *UserMap* as well as practitioners *PractiMap* and a mapping of blacklisted users *B_List*. We also define a set of modifiers: *onlyby()* which restricts the execution of certain functions to only the manager nodes and *onlyIf()* which requires that the *msg.sender* is not blacklisted. Furthermore, managers can designate roles

Algorithm 3 MultiSigMedRec Smart Contract

```
Require: Initiation of parameters: (Patient-add, Doctor-add,
   lastRecordID, timestamp, isValid);
Ensure: Setting up of modifiers: onlybySealer(); onlyby-
   Doc(); onlybyPatient(); onlybyManager();
   function createRecord(metadata of the EMR)
   set the parameters;
   isValid = false;
   end function
   function validateRecord(hash) onlybyDoc()
      if (EMRHash == hash) then
         isValide = true;
      else
         isValide = false;
      end if
   return isValide;
   end function
   function updateRecord(metadata of the EMR) onlybyPa-
   tient()
   set the parameters;
   isValid = false;
   end function
   function updateMIoTdata(RecordID, MIoTData) onlyby-
   Sealer()
   update MIoT data;
   end function
   function SendRecordURL(RecordID, Requestor_add,
   URL) onlybyManager()
      if (RecordAccess() == true) then
         decrypt data using secret key;
         encrypt data using requestor's public key;
         return URL hash pointer of the EMR;
      else
         access denied;
      end if
   end function
```

to users utilizing the *role_Assignment()* function, *ModifyStateOfRole()* is used to change the state of a permission, meanwhile, *CheckPermiOfRole()* verifies if a user has the required permission to access data. Moreover, we propose a registration smart contract, namely, *MultiSigMedRec* detailed in Algorithm 3. In which we define a structure to store the partial data of the EMRs on-chain and introduce a multisignature mechanism that gives a patient the right to upload his own EMR, but with the validation of the doctor who issued it to guarantee the credibility of the shared data. This was achieved by introducing a *bool* variable called *isValide*, initially set as *false* by default, and a function named *validateRecord()* which can only be executed by the EOA of the doctor who issued the record. Furthermore, the PHD generated from the MIoT devices are also updated in the global-blockchain by the ENs (i.e., authorized sealers) after local processing.

C. USE CASE SCENARIO

To further elaborate our framework idea, let us take the first scenario of a patient, upon his visit to a hospital some EMR will be generated, if both parties agree to share the data it will be registered by the patient in the global-blockchain with their respective signatures. The sequence diagram of the transaction flow to share health data is illustrated in Fig. 5, the first steps are the deployment and initiation phase in which the managers and edge nodes are responsible for the deployment of the smart contracts at the global and edge blockchain layers, as well as the registration of the verified users and practitioners on-chain by retrieving the data off-chain from the HCAs. The next step is for a patient to register his own EMR by interacting with the *MultiSigMedRec* smart contract using the *createRecord()* function that takes as inputs: the doctor EOA, the hash of the record, and a URL-like hash identifier of the record which will be saved on the blockchain pointing to the resource. As we have discussed previously, storing large quantities of text/media on the blockchain itself is not a good idea, rather a patient can use a decentralized file hosting service (e.g., IPFS) and get a hash that points to the uploaded encrypted data using his secret key, which is quite similar to a link to the files. Then, the record needs to be validated using the *validateRecord()* function that for ease and efficiency, simply compares the hashes of the records and ensures that the *msg.sender* address is the same as the doctor's EOA of the given record, meaning that the record gets to be validated *only and only by* the doctor who issued it in the first place using the *onlybyDoc()* modifier defined in the source code of the smart contract.

The second scenario to consider is of a patient monitoring his daily health signals from a smart home using a variety of MIoT devices, the monitored PHD will be sent to the local edge-mining blockchain with the signature of the owner, using the *MIoTdataUpdate()* function defined in the MIoT-Edge-Manager smart contract by interacting with the PN that functions as a gateway between the MIoT devices of the end-user layer and the local edge-mining layer. The data sent to the edge-mining pools would be processed locally to avoid ether overuse and only metadata would be sent to the global-blockchain by the authorized ENs within the local edge-mining pool, which are also lightweight nodes within the global-blockchain network, using the *updateMIoTdata()* function defined in the MultiSigMedRec smart contract which execution is restricted only to the authorized sealers using the *onlybySealer()* modifier.

Finally, the requestors can check the validity of any record at any given time, using the call function *CheckValidityOfRecord()* which doesn't require any fees, before requesting access to it by interacting with the RBAC smart contract. If the requestor is proven to have the required privilege and permissions to access the data, a manager node would decrypt and re-encrypt the requested record using the requestor's public key and a temporary URL hash pointer would be sent to his EOA. This is due to privacy concerns as all data are supposed to be encrypted off-chain with different keys, upon the request of access passed on the role of the requestor, a temporary hash pointer will be sent which is going to contain the encrypted data using the public key of the requestor, hence only his private key could decipher it. Then the requestor would be able to verify the credibility of the data after re-calculating the hash of the record and comparing it to the metadata stored on-chain before rewarding the patient.

V. IMPLEMENTATION AND EVALUATION

Nowadays, several blockchain platforms exist, including Bitcoin, Ethereum, Hyperledger, etc. In this paper, we have opted to use Ethereum for the implementation of our *EdgeMediChain* architecture as it has a large, global development community, it is completely open source, and it supports a variety of use cases such as smart contracts and decentralized applications (i.e., DApps) which can be built on top of the Ethereum blockchain, as it has been designed to be adaptable and flexible with Turing complete scripting. Furthermore, Ethereum is among the popular blockchain-based distributed platforms. Hence, a great deal of analysis and benchmarking of the platform in terms of performance and scalability is already provided by the literature [42]. Therefore, our implementation analysis deliberately ignores the evaluation of the Ethereum blockchain, but instead focuses on the impact that has the integration of new elements to our framework that are not among the components of the traditional Ethereum blockchain such as: the ENs and the IoT devices. Precisely, we evaluate how adding a set of local edge-mining pools to the blockchain architecture will influence the throughput of the system as well as the scalability of the ledge size vis-à-vis the huge amount of transactions generated from the MIoT devices.

A. TESTBED SETTINGS

Our proposed framework was deployed using Go-Ethereum (v 1.8.27) which is the Go-implementation of the Ethereum protocol, we used v 0.4.26+ for the solidity compiler and the source code of the proposed smart contracts can be found here [43]. The testbed is composed of five virtual machines (Ubuntu v 14.04.6) to emulate *EdgeMediChain* topology, running on a 2.4 GHz, Intel i-5, 4 GB 1600 MHz DDR3 laptop. One virtual machine is used to emulate the global-blockchain which uses the PoW consensus mechanism (i.e., Ethereum Ethash), while the remaining four are used for the parallel emulation of the local edge-mining pools using the PoA consensus mechanism (i.e., Clique), with the configuration specified in Table. 1. The distinction regarding the consensus mechanisms used is due to the different requirements in terms of latency and security (discussed in Subsubsection III-B1) while handling both EMRs and PHD as well as the nature of participants within each layer. In point of fact, getting access to the shared EMRs does not involve any emergencies in our proposed system, making it less critical vis-à-vis delays, as the whole point of the global-blockchain is to offer a secure and auditable sharing environment for the different entities involved. However, as the system involves a variety of entities making it less trustworthy there is a need to ensure that the mining of transactions is not going to be tempered with.

TABLE 1. Consensus protocols configuration.

Parameters	PoA	PoW
homesteadBlock	-	0
eip150Block	2	-
eip155Block	3	0
eip158Block	3	0
byzantiumBlock	4	-
period	0	-
epoch	30000	-
gasLimit	0x47b760	0x2100000
difficulty	0x1	10

Meanwhile, as the local edge-mining layer is assumed to be consortium and managing the huge amount of PHD generated from the MIoT devices is sensitive to delays and latency, PoA appears to be the best candidate as it has one distinctive feature: block sealing (or mining) is only performed by the verified signers. That's it, in order to validate a block there is no need to perform any hash mining; the only requirement is for the sealer to be included in the list of legitimate signers. Furthermore, in case of a malicious user getting added to the list of signers, or a compromised key/machine. The protocol ensures that given a list of N authorized signers, any signer is restricted to only seal one block out of every N/2, then the rest of the legitimate sealers can vote out the malicious or compromised node.

The experiment setup in each local edge-mining pool is as follows: one node is configured as a sealer running on full mode, with the number U of light mode users (i.e., patients) set to 1, 5, 10, 20, 40, and 100 users. Each user has a total of five MIoT devices and can send concurrent transactions (Tx) from each device. Users can send simultaneous Tx of function type *(f)* to the local blockchain without having to wait for a response. The amount of parallel transactions Tx is set to 20, 100, 200, 400, 800, and 2000 transactions. The transactions type *(f)* can be either *DeviceRegistration()*, *HashFirmwareUpdate()*, or *MIoTdataUpdate()*. Furthermore, the interactions between users and the blockchain platform are achieved with Node.js and all transactions are simulated with scripts using the web3.js library through JSONRPC call APIs [44].

B. EXPERIMENTAL RESULTS

In this subsection, we evaluate the performance of the proposed architecture in terms of the execution and sealing time, we also analyze the average throughput and ledger storage as well as the effect of the block period and number of sealers on the average latency. We should note that the results obtained represent the average of five independent runs and they are compared to an Ethereum-based traditional blockchain with one single mining pool running on the PoA consensus mechanism, referred to later as **PoAEB**. Finally, the deployment cost is also presented in terms of gas consumption.

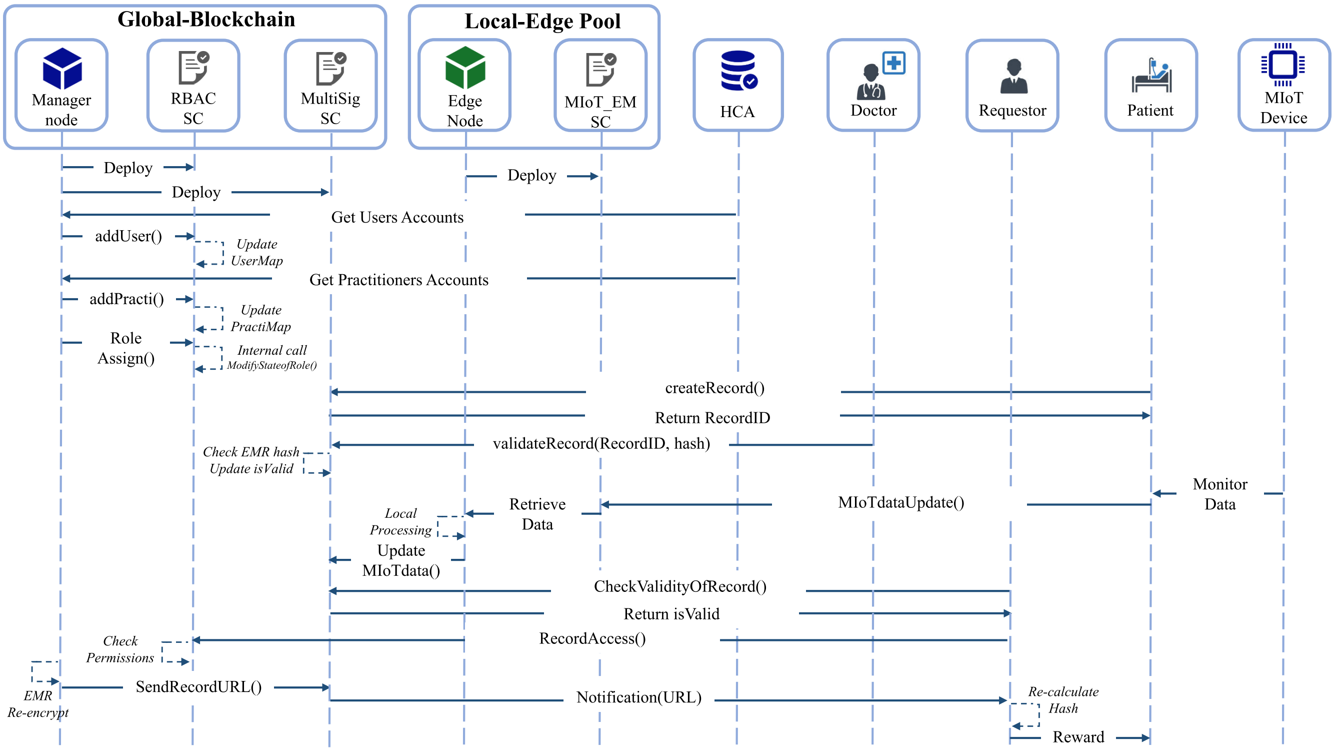


FIGURE 5. The overall transaction workflow of sharing and accessing health data.

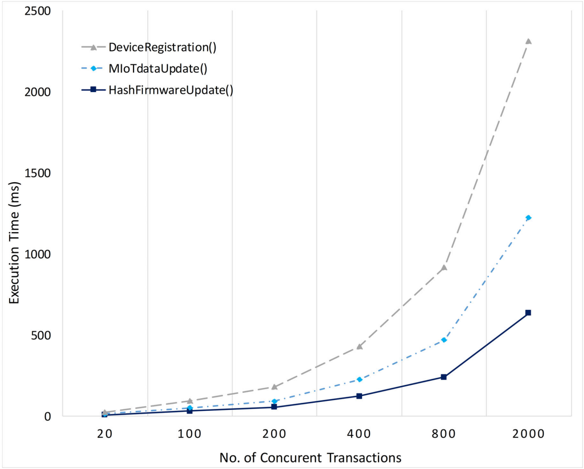


FIGURE 6. Comparison between the execution time of the different functions defined in the MIoT-Edge-Manager smart contract.

1) EXECUTION AND SEALING TIME

The execution time is defined as the required time needed for the sealers to group the unconfirmed transactions from the transaction pool, validate each one including running the associated smart contract code to fill the new block. We have investigated the change in terms of different numbers of Tx, the first observation to make is that the execution and sealing times both grow as the number of Tx in the evaluation set increases which is quite expected. However, it is important to highlight the gap between *EdgeMediChain* (EMC) and PoAEB which validates the effectiveness of our proposed solution. Fig. 6 represents a comparison between the execution time of different functions provided in the *MIoT-Edge-Manager* smart contract. The *DeviceRegistration()* function has a higher execution time and this is because it allocates an array for the first time to store all the parameters related to the registered MIoT device. Meanwhile, the *HashFirmwareUpdate()* and *MIoTdataUpdate()* functions have lower execution time as they simply modify the values in the storage associated with the smart contract. In Fig. 7 we compare the overall execution time of both EMC and PoAEB frameworks, we should note that a user with 5 MIoT devices will generate 5 concurrent Tx. Similarly, 10 patients will send 50 concurrent Tx to the network. According to Fig. 7, for 4 users each having 5 MIoT device, the execution time of 20 Tx takes *8.924ms* for EMC framework, while 100 concurrent Tx generated from 20 users require *32.135ms*. With a four times rise of concurrent Tx, the execution time of EMC increases slowly to finally reach with 400 users (generating 2000 concurrent Tx) *771.02ms* compared to *5.056s* for the PoAEB framework, achieving a reduction of nearly ∼ 84.75% in terms of execution time.

Meanwhile, the sealing time is defined as the time between the moment a block is committed after it has been validated to the moment it's chained. In Fig. 8 we compare the sealing time in both EMC and PoAEB frameworks, with different functions, against an increasing number of users similarly to the execution time. According to Fig. 8, for 4 users, the

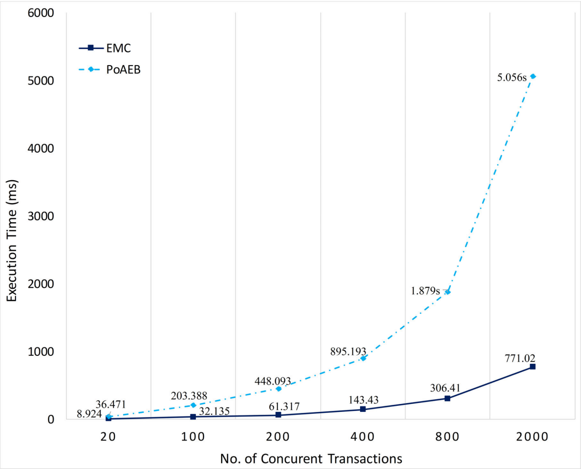


FIGURE 7. Comparison between the average execution time of EMC and PoAEB.

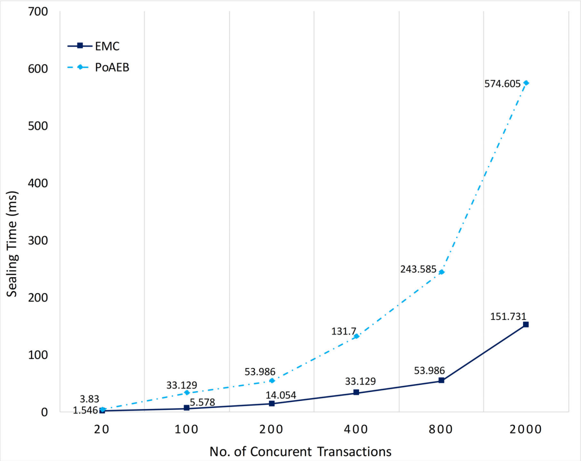


FIGURE 8. Comparison between the average sealing time of EMC and PoAEB.

sealing time of 20 Tx in EMC takes *1.546ms* and it increases slowly to finally reach *151.731ms* for 2000 concurrent Tx, compared to *574.605ms* for the PoAEB framework, achieving a reduction of nearly $\sim$ 73.59%.

2) THROUGHPUT

For the system throughput, we measured the number of successful Tx starting from the first transaction deployed until the last chained transaction. Fig. 9 highlights the plot of the average throughput with six experimental sets each with different loads. The results show that EMC has a higher throughput compared to PoAEB in all of the evaluation sets. We observe that the average throughput for EMC is maximum at 400 Tx nearly six times higher, while the lowest value is observed for 20 Tx. Furthermore, it can be observed that with different loads in terms of Tx, the variation of the average throughput in our proposed framework is relatively larger. This is because the PoAEB framework is already saturated and it reached the highest throughput it can achieve, meanwhile our proposed approach is able to achieve higher throughput before it starts to decline with the rise of the concurrent transactions the system has to handle.

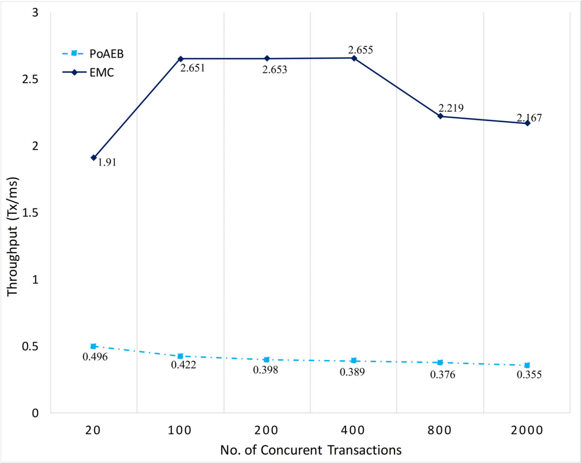


FIGURE 9. Comparison between the average throughput of EMC and PoAEB.

3) LEDGER SIZE

Each block of the growing shared ledger has a unique number associated with it and is linked to the previous block using a hash pointer to make a chain of data with the exception of the *genesis block*, which is created using the genesis state file or *genesis.json* in Geth. This file contains all the data needed for the generation of block 0, including the different accounts and their balances, the consensus protocol used, the chainID, the gasLimit, etc. We should note that Ethereum's block size is based on the complexity of contracts being run, known as the *gasLimit* per block which is a cap on both the processing and storage/bandwidth resources voted up or down by each miner where each one determines what *gasPrice* it is willing to accept. After a number of experiments, we observed that on average the block size was $\sim 38KB$ with a total of 64 Tx. This value is highly dependent on the nature of the transactions themselves, the gasLimit associated with the execution of a function of the smart contract, and the number of concurrent transactions in the transaction pool. We analyzed the effect of the increasing number of Tx and parallel processing on the scalability of the ledger. We note that the "period" in PoA was set to zero seconds, which allows us to study a higher processing rate of transactions and also stops the mining of empty blocks which will add up to the ledger size.

Fig. 10 represents the scalability of the ledger in both platforms, it highlights the cumulative memory against the increasing number of concurrent Tx as blocks are added in sequential order. From Fig. 10 we can see that the growth of PoAEB framework is higher compared to our proposed EMC framework as the transactions are divided among the different edge-mining pools, achieving a linear growth compared to an exponential one observed in the traditional approach, this is due to the fact that EMC allows for a parallel processing

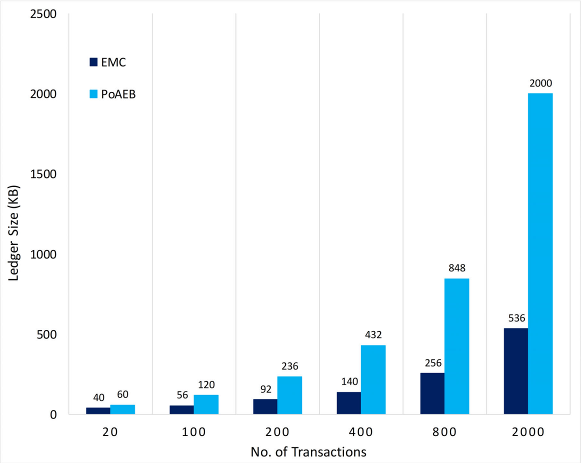


FIGURE 10. Comparison between the ledger storage scalability of EMC and PoAEB.

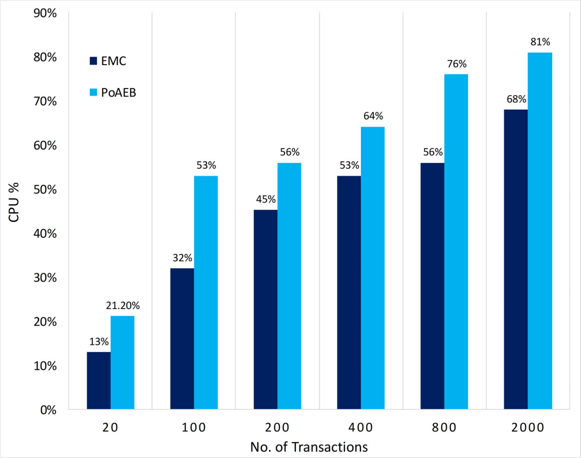


FIGURE 11. Scalability of CPU usage.

which requires lower memory resources within each mining pool.

4) CPU USAGE

The CPU measurements here are represented in percentages that dictates how much of the processor's capacities are being utilized by the mining of the transactions deployed. Figure. 11 represents the scalability of the CPU usage in both platforms, it highlights the percentage of the CPU resources used for the mining against the increasing number of concurrent Tx as blocks are being mined in sequential order.

From Fig. 11 it can be seen that the CPU usage of the PoAEB framework is higher compared to EMC as the transactions are divided among the different edge-mining pools, achieving a parallel processing and lower percentages of CPU resources required compared to the traditional approach.

5) BLOCK PERIOD

The PoA consensus mechanism (i.e., clique) leveraged in our framework is based on a single round block proposal in which one of the elected sealers (i.e., leader) is responsible for the creation of the new block that he broadcasts to all other sealers after signing it for the block to be chained right away, while dealing with forks later if they occur using the Ethereum GHOST protocol, therefore, decreasing the latency in terms of message rounds compared to PBFT consensus mechanisms and achieving better performance. The default block period specified for the protocol is 15 seconds which is the same duration in PoW (i.e., Ethash) to keep the network analogous to the main Ethereum network. However, in our implementation we have built a consortium private blockchain network, hence, the period can be set at any given value taking into account the trade-off between latency and the ledger size. Here, we evaluated the impact of the block period on the latency of the system, defined as the time between the first deployed transaction until the last chained block containing the last transaction in the network. We have set the number of sealers at two, whereas the period at different values: 5s, 10s, and 15s.

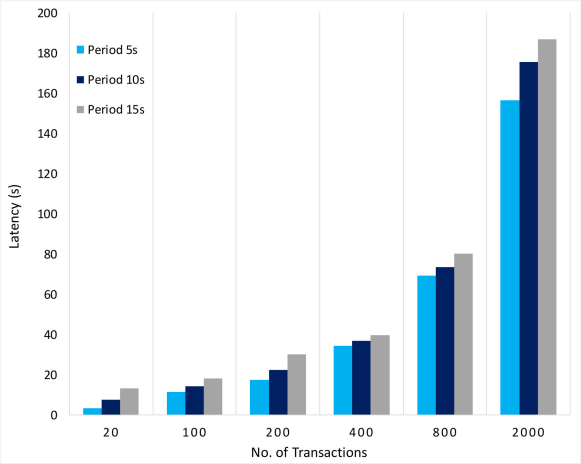


FIGURE 12. Latency of the system with different block periods.

From Fig. 12, we observe that the latency increases moderately with the increase of the block period which is expected as the sealers will have to wait for the defined period between two consecutive blocks creation. In order to obtain the optimum performance, the period has to be set in accordance with the frequency of the traffic within the system, as if transactions are not generated with a high frequency setting the block period at a lower value would speed up the chain, but would also increase the mining of empty blocks which will add up to the ledger storage as each empty block weights 1024 bytes. Meanwhile, if the period is set at a higher value with frequent transactions, this will affect the latency of the system by increasing the time needed for chaining all the generated transactions.

6) NUMBER OF SEALERS

At the beginning of each epoch a particular block used for transition is distributed among the nodes that contains the

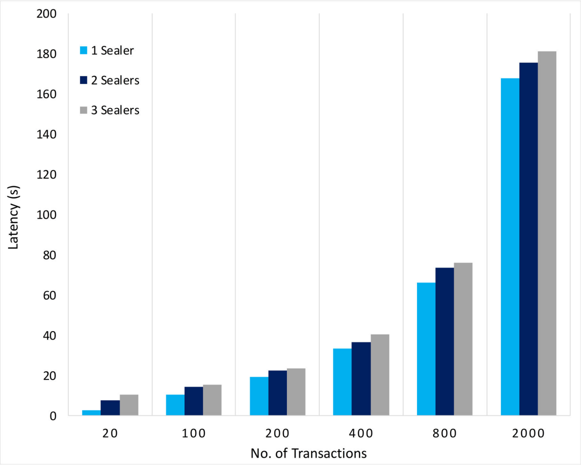


FIGURE 13. Latency of the system with different numbers of sealers.

list of authorized sealers allowed to create and sign blocks, it is also utilized by new sealers for network synchronization as the PoA consensus protocol allows to have a network of more than one sealer (i.e., authorized node) to create blocks with random delays, which enables the network to cope with sealers who were not able to send a block due to network asynchronization or Byzantine faults and guarantee consistency. Furthermore, the frequency of mining new blocks is limited by the N/2 rule, hence, in order to have full control over the chain the majority of Byzantine sealers is needed. Therefore, the more sealers the network has, the more resilient it is against attacks. Here, we evaluated the impact of the number of sealers on the latency with a block period set at 10s.

From Fig. 13, we observe that by increasing the number of sealers the latency has also increased slightly, however, this is predictable due to the delay introduced by the synchronization between sealers and the propagation of blocks before committing them, as introducing one sealer to the network increases the latency by 4s roughly. Hence, it is important to acknowledge that the more authorized nodes the network has, the more it would take for transactions to be chained, however, having a higher number of independent sealers would protect the network from being centralized or from the possibility of having a set of sealers fully controlling the chain.

7) COST AND FEASIBILITY

As for the authorization scheme we compiled and deployed our proposed RBAC and MultiSigMedRec smart contracts to the global-blockchain and we evaluated the deployment cost as well as the execution cost of some functions. During the phase of test, on March 2020, 1 ETH $\simeq$ 138 USD, and the minimal average value of gas was around $\sim$ 20 gwei. Actually, the gas value is disproportional to the time to mine a transaction, as a low gas value means that the transaction will take time to be validated by miners for they can ignore it and it is not processed with a higher priority. However, for our system and as we have discussed previously the requirements of sharing data, the validation time is not critical, hence, the minimum value is seen to be enough.

TABLE 2. Different execution fees of the proposed smart contracts. *(1 Gas = 20 gwei, 1 Eth = 10^9 gwei).*

Functions	Gas	Ether
Contracts creation	1413213	0.02826426
addUser()	47634	0.00095268
addPractitioner()	37939	0.00075878
role_Assignment()	24196	0.00048392
ModifyRole()	64715	0.0012943
createRecord()	152352	0.00304704
validateRecord()	29891	0.00059782
updateRecord()	44496	0.00088992
updateMIoTdata()	46103	0.00092206
RecordAccess()	24244	0.00048488
SendRecordURL()	26246	0.00052492

The execution costs of a variety of functions in the smart contract are listed in Table. 2, where the highest cost being attributed to the creation and deployment of the contracts with a value equal to $\sim$ 0.02 ETH. However, it is important to note that it's only paid for once to initialize the global-blockchain. Meanwhile, all the remaining functions have rather low fees, with the costs of *addPractitioner()* and *role_Assignment()* being around $\sim 0.0007 - 0.0004$ ETH respectively. Furthermore, during the testing phase using RemixID [45] we tried to lower the functions' fees by optimizing the code, for instance, we opted to use a *bytes32* type for roles rather than a *string* type as it consumes less gas compared to the latter. We also have paid extra attention to data structures and data-types in our proposed contract code by using mappings as far as possible instead of arrays (e.g., mapping of roles).

C. SECURITY ANALYSIS

Similarly to the work in [46], [47], this section presents some of the potential threats and attacks on the overall proposed framework and discusses how the proposed smart contracts achieve the security goals in terms of confidentiality, integrity, transparency, and privacy, while taking into account that constrained MIoT devices are also part of the system. Specifically, the proposed architecture mitigates the MIoT devices from the heavy workload of the authentication procedures as well as the computational tasks involved in interacting with the blockchain network as these tasks are actually carried out by the PNs.

1) CONFIDENTIALITY

In our proposed edge blockchain-based architecture this requirement is met by granting only legitimate access to the shared data. First, blockchain relieves the overall system from utilizing extravagant public key infrastructure, as unique 20-bytes Ethereum addresses are assigned instantaneously with no collision to any user. Second, restrictions in terms of access to the shared data are ensured by including modifiers in our source code of the smart contracts that will limit the access of different entities to only specific functions based on their privileges. For instance, we use the *PreConditions()* modifier that allows only the managers or practitioners having the right permission to execute certain functions such as the *role_Assignment()* or *Delegation()* functions. Hence, if another user tries to execute these functions, the execution will fail and trigger a revert exception.

2) INTEGRITY

Within an IoT ecosystem, integrity is highly needed to avoid any altering of the data as well as the credibility of the sender of the data. The proposed schemes ensure the security of the system, making it immune to Man-in-the-Middle attacks as every transaction triggering the execution of the smart contracts is signed using the node private key leveraging the ECDSA. Furthermore, the *ValidateHashFirmware()* function verifies the integrity of the firmware hash -which is the hash of the compiled firmware file within the MIoT device- to ensure it is complete, updated, and unaltered. In addition, the MultisigMedRec smart contract ensures a shared ownership between both the patient and the doctor, by allowing patients to upload their own medical records, but also requires the validation of the data by the doctor using the *isValid* parameter. Hence, only the authenticated medical records are considered valid by the data requestors, which guarantees the integrity of the shared health records. Moreover, setting the value of *Eip155Block* at 3 within the configuration of the *genesis file* prevents replay attacks. Last but not least, the proposed authentication scheme is resilient against DDoS attacks by assigning a value equal to 2 for the *Eip150Block* in the *genesis file*.

3) TRANSPARENCY

All functions executed in the smart contracts are logged in the Ethereum blockchain ledger. Hence, no entity will be able to execute any action without the others being aware of it and this is due to the immutable shared ledge feature of the blockchain technology. Furthermore, the decentralized ledger signifies that all transactions are going to be recorded in an identical manner across multiple nodes. In point of fact, the core of the blockchain transparency relies on saving all the records across a spread vast network for all users to see. That is also why blockchain is considered hacking-resistant as the record of transactions is indelible, it will be more difficult to commit fraud within the system.

4) PRIVACY

This is ensured by leveraging the Ethereum blockchain strong cryptography mechanisms as well as pseudo-anonymity. Precisely, each Ethereum user has a pair of key, a public key which anyone can see and is used to validate a signature associated with a transaction; and a private key which should be kept secret and only known to the owner as it is required to sign the transaction. However, even though any given transaction can be found in the ledger using a public key, connecting the public key to the owner of a private key or a real identity of a node is hardly possible. Furthermore, our proposed framework assumes that the shared data are encrypted and anonymized, which protects the privacy and the real identity of patients.

VI. RELATED WORK

In this section, we discuss the related work which we have divided into two folds, blockchain-based solutions dedicated for health data sharing and solutions addressing the scalability issue while combining blockchain with an IoT ecosystem.

A. BLOCKCHAIN FOR HEALTH DATA SHARING

Recently, blockchain has been widely used as the appropriate infrastructure for healthcare data sharing due to its powerful security features. For instance, the authors in [16] present MedRec, a decentralized EMRs management system, using the blockchain technology. In which the content of each block highlights data ownership as well as permissions shared by the network nodes. By utilizing Ethereum smart contracts, they associate EMRs with viewing permissions and data access instructions from external databases. However, the system relies heavily on the hospital centralized databases and does not provide the details of the permissions and AC policies. In MeDShare [17], the authors address the challenge of sharing health data between medical entities in a trust-less environment. The system provides data provenance, auditing, and control for the shared medical data in cloud repositories among big data entities. The framework leverages blockchain to effectively track the behavioral access to the data and revoke access to malicious entities upon the violation of permissions, however, the system does not provide any mechanism to ensure the credibility of the shared data. In [19] the authors propose the design of a medical information sharing platform based on blockchain using chaincodes running in docker containers, designed with the main goal of serving patients, hospitals, and third-party institutions. Users can interact with the platform via an application interface, while the process of accessing the information is recorded in the blockchain. The platform achieves the idea that data are values and by allowing access to it patients can get some rewards from institutions. In [26] the authors propose the design of a three modules framework to manage the access to patients' EMRs based on a role-based smart contract and they used off-chain storage to minimize the data stored on-chain. However, the system is not patient-centric in a sense that patients only have the view right to their own EMRs, furthermore, the proposed smart contract specifies the names of the patients which might be a violation of their privacy and confidentiality according to the HIPAA. In [27], the authors propose the design of a blockchain-based system to manage the access to the EMRs by leveraging smart contracts, they also proposed a new consensus mechanism to create blocks which is built on the PoA protocol, the mechanism utilizes a degree of like-hood to create new blocks. However, the authors did not detail how they implemented their scheme with the PoA mechanism as in the latter the node creating a block is also the one to sign it. In [28], the authors leveraged a hyperledger-based permissioned blockchain system to manage the emergency access to patients' EMR using pre-defined rules in case they are in a situation where they are unable to give consent, however, the algorithms proposed for the registration provide the names of patients and doctors, which might violate their privacy as all transactions within the blockchain are visible to all nodes part of the network. Similarly, the work in [48] proposed to use the hyperledger fabric for a multi-layered architecture to manage health information they also included IoT devices and body sensors in their system, however, these devices are under the supervision of the health providers rather than the patients, not to mention that the scalability issue these devices would indulge was seldom investigated. Meanwhile, In order to ensure the validity of the shared EMRs within a system encompassing multiple non-trusted authorities, the authors in [49] introduced an attribute-based signature with blockchain that allows for patients the endorsement of a message without the disclosure of any further private details. In [50] the authors leveraged cloud-based storage with blockchain to manage the access to EMRs, meanwhile, security and privacy were preserved by utilizing proxy-based re-encryption and a designed proof of authorization (PoAuth) consensus mechanism. However, the proposed approach does not give full ownership of the records to the patients as these ones are still under the management of the healthcare providers and they are the ones responsible for uploading the EMRs to the cloud. Similarly, the work in [51] combined cloud storage, attribute-based and searchable symmetric encryption with blockchain and smart contracts to meet the privacy needs for a secure, reliable and restricted access control over the shared EMRs, however, the authors only provided the time required for the files' encryption without introducing the overhead of the blockchain technology and mining.

B. BLOCKCHAIN AND IoT

Several solutions in the literature have been proposed to tackle the scalability issue of blockchain while merging it with IoT. For instance, Novo proposes to leverage blockchain as the building block of a decentralized access management system for IoT devices in [47], where AC information is stored and distributed on-chain. The IoT ecosystem is connected to the blockchain network by interacting with a gateway called management hub, allowing any IoT node to query other blockchain nodes without endorsement or a well-defined device identification mechanism. Hence, the paper eludes this verification, which makes the system substantially insecure in case of a malicious manager. In [52] the authors propose a blockchain based framework for IoT to handle inter and intra-organizational transactions in which all IoT devices are supposed to be associated with their own respective organization. Furthermore, instead of utilizing a single peer, part of the global-blockchain network, they propose a local peer (Lpeer) structure. However, the whole architecture relies on a single local peer (i.e., Lpeer0) and the secondary peers are only activated if Lpeer0 is out of service, which might not be enough to handle the growing amount of transactions generated and most importantly does not take into account the case in which Lpeer0 is compromised by an attacker. In [53] an IoT lightweight, scalable blockchain (LSB) solution is presented in which the authors introduce a block manager (BM) which plays the role of a centralized manager similar to a hub. However, the synchronization issue between the local BM and the overlay BM in this work was hardly inspected. In [54] the authors propose a mobile edge computing (MEC) enabled blockchain network in which mobile users (e.g., mobile or IoT devices) can offload the task of mining blocks to the deployed edge servers at the MEC service provider level due to their constraints in terms of computation and storage resources. They also studied the pricing for the provided mining services for which they proposed a Stackelberg game and defined the conditions of the Nash equilibrium. However, the system is still vulnerable as transactions can still be intercepted while being offloaded to the MEC provider, furthermore, DDoS can easily be launched as there is no mechanism to verify the mobile users. In [55] the authors combined the blockchain technology with the fog computing paradigm and proposed an architecture dedicated for IoT applications in smart cities, which secures data using encryption and guarantees a reduced latency and energy consumption, however, the details of the blockchain ledger storage, mining nodes, and consensus mechanism were hardly inspected.

C. SYSTEM COMPARISON

The already existing work related to sharing EMRs using blockchain is enormous. However, few solutions focus on sharing the huge amount of PHD generated from wearable MIoT devices, which can play a crucial role in the overall health services delivery. In this work, we proposed a decentralized MIoT authentication scheme which is different than the already existing blockchain-based schemes in the literature as it does not only authenticate the users part of the system, but it is also aimed towards solving the issue of counterfeit devices and firmware updates which are two crucial aspects of MIoT, as medical devices are getting more attention recently they are also becoming the target of hackers and are not getting the required and needed security management. In addition, the traditional centralized approaches are enable to cope with the huge amount of data let alone their vulnerabilities such as data breaches and DDoS attacks. Hence, our proposed edge blockchain-based architecture ensures a decentralized monitoring of all devices and protects the system from non-authentic devices by keeping patients aware of new patches and updates needed to guarantee the security of their devices. Furthermore, the mechanisms used for the authorization procedure in the existing works (discussed in Subsection VI-A) are hardly detailed. Meanwhile, our proposed RBAC smart contract constitutes a dynamic, secure, open, transparent and decentralized authorization scheme to manage the access control to the shared health data as it allows for a flexible and fluid management, while still being

TABLE 3. Blockchain-based HIE systems comparison.

Metrics	[16]	[19]	[22]	[26]	[27]	[28]	[48]	[50]	[51]	EMC
Platform	Ethereum	Hyperledger	Hyperledger	Ethereum	Ethereum	Hyperledger	Hyperledger	Ethereum	Ethereum	Ethereum
Consensus	PoW	PBFT	PBFT	PoW	PoA	PBFT	PBFT	PoAuth	-	PoW/PoA
EMRs and PHD	×	✓	×	×	×	×	✓	×	×	✓
Edge-based	×	×	×	×	×	×	×	×	×	✓
Dynamic AC	×	×	✓	✓	✓	✓	×	✓	✓	✓
Scalability and IoT	×	×	×	×	×	×	×	×	×	✓
Data anonymity	✓	×	×	×	×	×	×	×	×	✓
Patient-centric	×	✓	✓	×	×	✓	✓	×	×	✓

compliant with the HIPAA to preserve the real identity of participants and guarantee privacy, as unlike some of the solutions provided in the literature our scheme does not require to fill real names at the registration phase on-chain. Finally, the edge-blockchain sharding architecture designed and implemented enables achieving a higher throughput compared to the traditional blockchain approach. The performance evaluation does not only show the effectiveness of the solution in terms of throughput, but also in terms of ledger storage and CPU usage which take into account the constrained ENs resources and enables processing the enormous volume of transactions expected to be generated from the MIoT devices with a higher speed, which meet the required application needs. Furthermore, the proposed sharding implementation ensures the security of the network as mining is performed only by authorized sealers who are also responsible for the cross-shards communication with the global-blockchain. As a summary, Table. 3 provides a qualitative comparison between our proposed framework and existing works in terms of different metrics.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed *EdgeMediChain* an authentication and authorization framework for sharing health data, including both the EMRs and PHD generated from MIoT devices by leveraging both edge computing to ensure scalability and the blockchain technology for security. We implemented an Ethereum-based prototype to test the proposed smart contract-based schemes, which are responsible for managing the interactions between all the entities of the system in terms of uploading and sharing patients' health data. The deployment tests show the efficiency of the proposed architecture by ensuring a lower execution time and CPU usage, higher throughput and a linear ledger size scalability compared to a traditional blockchain. We also evaluated the impact of the block period and the number of sealers within each pool on the latency of the system as well as the feasibility of the proposed solution in terms of transactions' cost and its resilience and effectiveness in terms of confidentiality, data integrity, transparency and privacy.

The experiments conducted in terms of generation of transactions were done using simulations rather than real MIoT devices, however, in the future we aim to analyze real transactions between the IoT devices, the PNs, and the mining pools deployed. Besides, for privacy concerns we assumed that the shared data is anonymized by omitting certain PII, nonetheless, other variations of k-anonymity, t-closeness, and l-diversity techniques can be implemented to guarantee that the data would not be re-identified by matching it to other open available databases. Furthermore, with artificial intelligence and machine learning emerging as revolutionary tools, a healthcare blockchain-edge infrastructure would undoubtedly allow for much more secure, effective, and rich medical research. As with a vast, standardized, anonymous, and temper proof data storage, research institutions can conduct significant clinical trials and prospective analyses, in this regards we aim to propose algorithms dedicated for training the system to proactively predict intelligent insights for better real-time decision-making and diagnosis.

ACKNOWLEDGMENT

The authors would like to thank Dr. K. Sharif, Prof. F. Li as well as the anonymous reviewers for their valuable comments and insightful suggestions that enhanced the clarity of the article.

REFERENCES

[1] (2016). World Health Organization (WHO), Geneva, Switzerland. *Global Report on Diabetes*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.who.int/diabetes/global-report/en/

[2] B. E. Dixon and C. M. Cusack, "Measuring the value of health information exchange," in *Health Information Exchange*. Amsterdam, The Netherlands: Elsevier, 2016, ch. 15, pp. 231–248.

[3] K. Kupferschmidt and J. Cohen, "WHO launches global megatrial of the four most promising coronavirus treatments," *Sci. Mag.*, Mar. 2020. Accessed: Jun. 11, 2020. [Online]. Available: https://www.sciencemag.org/news/2020/03/who-launches-global-megatrial-four-most-promising-coronavirus-treatments, doi: 10.1126/science.abb8497.

[4] S. M. Riazul Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[5] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[6] R. Lin, Z. Ye, H. Wang, and B. Wu, "Chronic diseases and health monitoring big data: A survey," *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 275–288, 2018.

[7] F. S. Collins and H. Varmus, "A new initiative on precision medicine," *New England J. Med.*, vol. 372, no. 9, pp. 793–795, Feb. 2015.

[8] P. Wu, C. Cheng, C. D. Kaddi, J. Venugopalan, R. Hoffman, and M. D. Wang, "–Omic and electronic health record big data analytics for precision medicine," *IEEE Rev. Biomed. Eng.*, vol. 64, no. 2, pp. 263–273, Feb. 2016.

[9] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: Taxonomy and survey," *Softw. Pract. Exper.*, vol. 44, no. 3, pp. 369–390, Mar. 2014.

[10] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, ''Fog computing in healthcare—A review and discussion,'' *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[11] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, ''An edge computing based smart healthcare framework for resource management,'' *Sensors*, vol. 18, no. 12, p. 4307, Dec. 2018.

[12] K. J. Madukwe, I. J. F. Ezika, and O. N. Iloanusi, ''Leveraging edge analysis for Internet of Things based healthcare solutions,'' in *Proc. IEEE 3rd Int. Conf. Electro-Technol. Nat. Develop. (NIGERCON)*, Owerri, Nigeria, Nov. 2017, pp. 720–725.

[13] D. Singh, G. Tripathi, A. M. Alberti, and A. Jara, ''Semantic edge computing and IoT architecture for military health services in battlefield,'' in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2017, pp. 185–190.

[14] N. Tariq, M. Asim, F. Al-Obeidat, M. Zubair Farooqi, T. Baker, M. Hammoudeh, and I. Ghafir, ''The security of big data in fog-enabled IoT applications including blockchain: A survey,'' *Sensors*, vol. 19, no. 8, p. 1788, Apr. 2019.

[15] J. J. Hathaliya and S. Tanwar, ''An exhaustive survey on security and privacy issues in healthcare 4.0,'' *Comput. Commun.*, vol. 153, pp. 311–335, Mar. 2020.

[16] 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)*, Vienna, Austria, Aug. 2016, pp. 25–30.

[17] Q. 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.

[18] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, ''Blockchain-based personal health data sharing system using cloud storage,'' in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Ostrava, Czech Republic, Sep. 2018, pp. 1–6.

[19] J. Chen, X. Ma, M. Du, and Z. Wang, ''A blockchain application for medical information sharing,'' in *Proc. IEEE Int. Symp. Innov. Entrepreneurship (TEMS-ISIE)*, Beijing, China, Mar./Apr. 2018, pp. 1–7.

[20] A. Theodouli, S. Arakliotis, K. Moschou, K. Votis, and D. Tzovaras, ''On the design of a blockchain-based system to facilitate healthcare data sharing,'' in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./ 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, New York, NY, USA, Aug. 2018, pp. 1374–1379.

[21] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, ''BHEEM: A blockchain-based framework for securing electronic health records,'' in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[22] S. Tanwar, K. Parekh, and R. Evans, ''Blockchain-based electronic healthcare record system for healthcare 4.0 applications,'' *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102407.

[23] P. Bhattacharya, S. Tanwar, U. Bodke, S. Tyagi, and N. Kumar, ''BinDaaS: Blockchain-based deep-learning as-a-Service in healthcare 4.0 applications,'' *IEEE Trans. Netw. Sci. Eng.*, early access, Dec. 2020, doi: 10.1109/TNSE.2019.2961932.

[24] *Sharding Roadmap*. Accessed: Jun. 15, 2020. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-roadmap

[25] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, ''A survey on the scalability of blockchain systems,'' *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, Sep. 2019.

[26] A. Shahnaz, U. Qamar, and A. Khalid, ''Using blockchain for electronic health records,'' *IEEE Access*, vol. 7, pp. 147782–147795, 2019.

[27] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, ''MedChain: A design of blockchain-based system for medical records access and permissions management,'' *IEEE Access*, vol. 7, pp. 164595–164613, 2019.

[28] A. R. Rajput, Q. Li, M. Taleby 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.

[29] R. Akkaoui, X. Hei, C. Guo, and W. Cheng, ''RBAC-HDE: On the design of a role-based access control with smart contract for healthcare data exchange,'' in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, Yilan, Taiwan, May 2019.

[30] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[31] *Solidity*. Accessed: Jun. 15, 2020. [Online]. Available: https://solidity.readthedocs.io/en/develop/

[32] G. Wood. Ethereum: A Secure Decentralized Generalized Transaction Ledger Byzantium Version. Yellow Paper. 2019. Accessed: Jun. 15, 2020. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[33] *Coin Market Cap*. Accessed: Jun. 15, 2020. [Online]. Available: https://coinmarketcap.com/currencies/ethereum/

[34] N. Szabo. Accessed: Jun. 15, 2020. *Formalizing and Securing Relationships on Public Networks*. [Online]. Available: https://nakamotoinstitute.org/formalizing-securing-relationships/

[35] Summary of the HIPAA Pivacy Rules. *Office of Civil Rights (OCR)*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.hhs.gov/sites/default/files/privacysummary.pdf?language=es

[36] E. C. O'Brien, A. M. Rodriguez, H.-C. Kum, L. E. Schanberg, M. Fitz-Randolph, S. M. O'Brien, and S. Setoguchi, ''Patient perspectives on the linkage of health data for research: Insights from an online patient community questionnaire,'' *Int. J. Med. Informat.*, vol. 127, pp. 9–17, Jul. 2019.

[37] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, ''A survey on security and privacy issues of bitcoin,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 2018.

[38] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P. K. Singh, and W.-C. Hong, ''Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward,'' *IEEE Access*, vol. 8, pp. 474–488, 2020.

[39] *The InterPlanetary File System*. Accessed: Jun. 15, 2020. [Online]. Available: https://github.com/ipfs/ipfs

[40] N. Davé, ''Cyberattacks on medical devices are on the rise-and manufacturers must respond,'' *Human OS*, Dec. 2019. Accessed: Jun. 15, 2020. [Online]. Available: https://spectrum.ieee.org/the-human-os/biomedical/devices/cyber-attacks-on-medical-devices-are-on-the-riseand-manufacturers-must-respond

[41] ANSI INCITS 359-2004 American National Standard for Information Technology. *Role-Based Access Control*. Accessed: Jun. 15, 2020. [Online]. Available: https://profsandhu.com/journals/tissec/ANSI+INCITS+359-2004.pdf

[42] *Ethereum Benchmarks*. Accessed: Jun. 15, 2020. [Online]. Available: https://github.com/ethereum/wiki/wiki/Benchmarks

[43] *EdgeMediChain Project—Source Code of the Smart Contracts*. Accessed: Jun. 15, 2020. [Online]. Available: http://cloud.eic.hust.edu.cn:8084/~raifa/bc-edge-hie.html

[44] *web3.js—Ethereum JavaScript API*. Accessed: Jun. 15, 2020. [Online]. Available: https://github.com/ethereum/web3.js

[45] *Remix-IDE*. Accessed: Jun. 15, 2020. [Online]. Available: https://github.com/ethereum/remix-ide

[46] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, ''A traceable blockchain-based access authentication system with privacy preservation in VANETs,'' *IEEE Access*, vol. 7, pp. 117716–117726, 2019.

[47] O. Novo, ''Blockchain meets IoT: An architecture for scalable access management in IoT,'' *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[48] M. Zghaibeh, U. Farooq, N. U. Hasan, and I. Baig, ''SHealth: A blockchain-based health system with smart contracts capabilities,'' *IEEE Access*, vol. 8, pp. 70030–70043, 2020.

[49] 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.

[50] Y. Wang, A. Zhang, P. Zhang, and H. Wang, ''Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain,'' *IEEE Access*, vol. 7, pp. 136704–136719, 2019.

[51] S. Wang, D. Zhang, and Y. Zhang, ''Blockchain-based personal health records sharing scheme with data integrity verifiable,'' *IEEE Access*, vol. 7, pp. 102887–102901, 2019.

[52] S. Biswas, K. Sharif, F. Li, B. Nour, and Y. Wang, ''A scalable blockchain framework for secure transactions in IoT,'' *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4650–4659, Jun. 2019.

[53] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, ''LSB: A lightweight scalable blockchain for IoT security and anonymity,'' *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019.

[54] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, ''When mobile blockchain meets edge computing,'' *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[55] P. Singh, A. Nayyar, A. Kaur, and U. Ghosh, ''Blockchain and fog based architecture for Internet of everything in smart cities,'' *Future Internet*, vol. 12, no. 4, p. 61, Mar. 2020.

RAIFA AKKAOUI received the M.Sc. degree in information and communication technologies from the National Institute of Posts and Telecommunications, Rabat, Morocco, in 2016. She is currently pursuing the Ph.D. degree in information and communication engineering with the Huazhong University of Science and Technology, Wuhan, China. Her research interests include blockchain, edge computing, the Internet of Things, network security, and game theory.

XIAOJUN HEI (Member, IEEE) received the B.Eng. degree in information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1998, the M.Phil. degree in electrical and electronic engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2000, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2008. Since 2008, he has been an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. He is a coauthor, together with Y. Liu and K. W. Ross, of the Best Paper in Multimedia Communications for 2008 awarded by the Multimedia Communications Technical Committee of the IEEE Communications Society. His current research interests include edge networking, intelligent healthcare, and embedded network systems.

WENQING CHENG (Member, IEEE) received the B.S. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985 and 2005, respectively. She is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. Her current research interests include information systems and e-Learning applications.

• • •