This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Access Control and Privacy-Preserving Blockchain-Based System for Diseases Management

Kebira Azbeg, Ouail Ouchetto, and Said Jai Andaloussi

***Abstract*— In many developing countries, the healthcare sector is facing several challenges, mainly due to the lack of personal, institutions, and medications in public health systems. Over the past decade, information and communication technology has proved its ability to improve medical quality, reduce costs, and promote data security. Developing countries can exploit these technologies to improve the healthcare process and ensure remote health monitoring, especially in rural areas. The Internet of Things and smart medical devices are widely used to provide remote patient monitoring. Current systems are based on centralized communication with cloud servers. However, this architecture increases several security and privacy risks. The adoption of a distributed architecture is required to overcome these issues. In this article, we describe a Blockchain-based system for securing Internet-of-Things (IoT) healthcare devices. In addition to data encryption, we propose to use Blockchain technology to enhance security and privacy in healthcare systems. The system is intended to allow remote patient monitoring, particularly for chronic diseases that necessitate regular monitoring. Three important characteristics were taken into account: security, scalability, and processing time. The security concerns are ensured by using the re-encryption proxy in conjunction with Blockchain to encrypt data and control access to it. To ensure Blockchain scalability, data are stored in an InterPlanetary file system (IPFS) off-chain database. We use an Ethereum Blockchain based on proof of authority (PoA) to speed up the data storage. In comparison to existing methods, the experimental system has shown a significant improvement in the security of healthcare systems.**

***Index Terms*— Blockchain, healthcare, information and communication technology (ICT), Internet of Things (IoT), InterPlanetary file system (IPFS), proxy re-encryption, remote patient monitoring, security.**

I. Introduction

HEALTHCARE sector in many developing countries is facing several challenges due to cultural, cost, and economic conditions. These challenges include the lack of personal, institutions, and medications in public health systems. To ensure adequate coverage with primary care interventions, the World Health Organization (WHO) estimates that at least 2.5 medical workers per 1000 people are required. Whereas, in lower-middle-income countries, it is only 0.8 per 1000 people in 2017 [1]. In addition, these countries have large rural areas that suffer from poor access to healthcare services. For example, hospitals and clinics may be far away. Furthermore, chronic health problems, such as diabetes, are increasing and often undiagnosed in these countries [2].

WHO defines chronic diseases as illnesses that are of long duration and generally slow progression [3]. According to the same organization, the leading causes of death and disability in the world are chronic diseases. There are four major chronic diseases, namely, cancer, cardiovascular disease (CVD), chronic obstructive pulmonary disease, and type 2 diabetes. For example, diabetes caused 6.7 million deaths in 2021 [4]. Approximately 537 million adults are living with diabetes worldwide. In addition, three in four adults with diabetes live in low- and middle-income countries.

Over time, uncontrolled chronic diseases can develop and lead to several complications and increase the risk of death. However, patients suffering from a chronic disease can live healthily and have a near-normal life if the disease is early detected and well managed. This kind of disease requires regular checkups and serious self-care. Thus, it can be monitored and we can then prevent its progress. However, the overall glycemic control and management of diabetes, in developed countries, is unsatisfactory, due to many factors such as lack of awareness, drug, and insulin costs. As a result, a significant number of patients fail to meet their treatment objectives.

Nowadays, technological advancement such as IoT, smart medical devices, and mobile applications can support daily healthcare activities to control chronic diseases and provide remote patient monitoring. These technologies are integrated into healthcare systems creating what is called the e-healthcare system.

Electronic healthcare (e-healthcare) is related to the adoption of different information and communication technologies (ICTs) in healthcare systems. It can be defined as “the infrastructure and health applications that use the technologies of digital communication networks of multimedia data, primarily the Internet. In simplified form, these terms are used to refer to the Internet in healthcare” [5]. The main benefits of such a system are the improvement of patient management, diagnosis, and follow-up.

Chronic diseases are one of the problems facing healthcare systems, in terms of sustainability of management. The development of ICT applications for healthcare systems is facilitating various tasks related to the control of these diseases. This enables doctors to assist patients in different phases, from

Manuscript received 25 September 2021; revised 6 April 2022 and 20 June 2022; accepted 24 June 2022. *(Corresponding author: Kebira Azbeg.)*

The authors are with the Computer Science and Systems Laboratory, Department of Mathematics and Computer Science, Faculty of Sciences Ain Chock, Hassan II University of Casablanca, Casablanca 20000, Morocco (e-mail: kebira.azbeg-etu@etu.univh2c.ma; ouail.ouchetto@etude.univcasa.ma; said.jaiandaloussi@etude.univcasa.ma).

Digital Object Identifier 10.1109/TCSS.2022.3186945

2329-924X © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

[Type here]

diagnosis to monitoring and treatment of chronic diseases. The IoT is a sophisticated technology that can provide an infrastructure to enhance the development of e-healthcare systems. It can provide, for example, tools (particular sensors) to collect vital signs from the patient, such as blood pressure, glucose level, and heart rate [6]. These parameters are then used to provide remote monitoring for patients. As such, e-healthcare platforms can help physicians automate the process of information transmitted by medical sensors to detect and report risk situations [7].

In general, IoT devices are referring to devices with embedded sensors that can connect over the Internet and exchange information. Every year, thousands of devices join the Internet to store their data on centralized servers. The device's authentication, authorization, and connection are relaying on a client/server system. This architecture raises several questions regarding security and privacy such as single point of failure [8], authentication, and access control [9]. Thus, the necessity of moving to a decentralized system, especially in the healthcare sector, devices manage sensitive and personal data, which requires a high security level. The Blockchain, which secures Bitcoin transactions since 2008 [10], could add a security layer to the IoT systems.

In this article, we present a Blockchain-based system to secure the IoT healthcare devices and the interaction between them at different levels. Our solution aims to collect, store, and share patient health data with healthcare teams in a secure manner while preserving patients' privacy. The distributed nature of Blockchain technology can make the system more robust to a single point of failure. In addition, the use of smart contracts can allow secure authentication for devices and handle access control to data. The system architecture is based on a set of technologies, namely, IoT healthcare devices, Blockchain technology, Smart contracts, InterPlanetary file system (IPFS), and proxy re-encryption.

The idea behind our solution is to propose a complete system, from the collection of health data to the analysis of these data. Data security is provided by the use of Blockchain technology and smart contracts, and data analysis will be provided through the use of machine learning algorithms. The main objectives of this work can be summarized as follows.

1) *Provide health data collection:* Using different IoT medical devices.
2) *Ensure efficient data sharing while protecting patients' privacy:* By adopting smart contracts and proxy re-encryption to handle access control.
3) *Provide a reliable storage:* By using IPFS to protect data integrity and ensure nonrepudiation.
4) *Ensure data traceability:* Blockchain provides a full history of all events and operations made on the data.
5) Develop a mobile DApp for patients and physicians to monitor patients and track their health situation.

The rest of this article is organized as follows. Section II highlights the security requirements and privacy rules for healthcare. Section III defines some preliminaries concerning Blockchain and IPFS technologies. Section IV describes the system design of the proposed approach. Section V illustrates the system working flow and a case study based on diabetes management. Section VI gives information about the system implementation. Section VII presents the security analysis of the proposed system. Section VIII presents the existing works and performs a comparison between these works and the proposed approach. Finally, in Section IX, we conclude the article and introduce some future works.

II. Healthcare Data Security Requirements and Privacy Rules

Due to the sensitive nature of the data handled in a healthcare application, privacy and security are primordial concerns.

A. Healthcare Data Security and Privacy Rules

To improve the efficiency and effectiveness of the healthcare system and to protect sensitive patient data from being disclosed, many regulatory standards were created. One of these regulators is the United States federal law Health Insurance Portability and Accountability Act (HIPAA) which was passed by the U.S. Congress in 1996 [11]. Some principal measures can be built in to secure Electronic Health Records (EHRs) systems [12], namely, access control, data encryption, and audit trail.

In addition, the HIPAA privacy rule created national standards to protect patient data [13]. It provides different rules to ensure the privacy [14] and security [15] of health information when it is transferred, received, handled, or shared. Among these rules, HIPAA.

1) Gives patients more control over their data. They have the ability to choose how their data will be used.
2) Sets limits on the use of data.
3) Establishes safeguards that healthcare providers must follow to ensure data security.
4) Imposes penalties on violators who violate patients' privacy rights.

B. Security Requirements in IoT-Based Healthcare Systems

Despite their important role in providing healthcare services with improved quality, e-healthcare systems raise new security issues for patients. These concerns are related to the technologies used in e-healthcare systems. For example, IoT, cloud storage, and communication links are suffering from different issues, both individually and when combined to form these systems. Patient data, which is extremely confidential and private, can be compromised and hacked at any point, from sensors to cloud storage. Security threats and cyber-attacks on e-healthcare systems can come from anywhere and at any level. Therefore, it is necessary to use efficient security mechanisms to protect e-healthcare systems from all threats [16].

The use of IoT in the medical sector can be classified into two categories: services (ambient-assisted living and Internet of m-health) and applications (glucose levels sensing and blood pressure monitoring). However, a number of security and privacy requirements should be considered for IoT-healthcare applications [17], including confidentiality,

[Type here]

integrity, authentication, availability, nonrepudiation, authorization, and privacy. The protection of personal data from others and malicious devices should be guaranteed at different levels: device level, during communication, at the storage, and processing levels [18]. Furthermore, several security challenges are facing the integration of IoT in healthcare, especially because of limitations in terms of computation, memory, and energy. Moreover, the mobility nature of IoT, which enables devices to be connected to the Internet through different networks, can affect its security.

III. Preliminaries

A. Blockchain Technology Overview

Blockchain is a technology that ensures data storage and transmission, is transparent, secure, and works without a central control entity. In addition, Blockchain is a platform to execute smart contracts that are embedded procedures intended to be automatically executed when the predefined conditions are satisfied. Due to its various characteristics, such as being secured, decentralized, and distributed, Blockchain has become extremely popular in practically all domains. There is no need for a central authority to govern the network because it is decentralized. To obtain agreement among nodes, data are archived via a consensus process. The ledger is distributed, and it is maintained by all of the network's nodes. Blockchain security is achieved by many concepts, including cryptography, consensus mechanisms, immutability, replication, and data traceability.

There are three forms of Blockchain: public, private, and consortium. The distinction is due to the consensus process and the visibility of data. Everyone can participate in the consensus process, have access to the ledger, and add data to it in a public Blockchain. However, in a private Blockchain, both access to the ledger and participation in the consensus process require permission from the owning organization. In a consortium Blockchain, both variants are grouped together. The ledger can be accessed in a public or private manner. However, instead of a single organization, numerous organizations can govern the consensus process.

B. Ethereum Platform

Ethereum [19] is a decentralized and distributed computing platform based on Blockchain. It was created by Vitalik Buterin in 2014 inspired by the Bitcoin cryptocurrency. Ethereum is based on the elliptic curve digital signature algorithm (ECDSA) standard. In the elliptic curve cryptography, the discrete logarithm problem is used for key generation. Ethereum uses the same elliptic curve, called secp256k1, as Bitcoin [20]. The principal goal of Ethereum is to provide Ethereum virtual machines (EVMs) that can run "Smart contract" programs.

C. Smart Contracts

Smart contracts are programs that are deployed and stored on the Ethereum Blockchain network to be executed by the EVM. These programs translate a contractual logic into a set of rules and regulations. When the predefined requirements are met, they can be self-executed without the involvement of a third party. A smart contract can store data, send transactions, trade values, or assets, and interact with other smart contracts. Ethereum also serves as a platform for developing and deploying decentralized apps (DApps). The backend of a DApp is built using Blockchain smart contracts. As a result, rather than being implemented on a central server, the back-end is built using smart contracts and runs on the Blockchain network.

D. Proof of Authority

Proof of authority (PoA) is a consensus algorithm in that only predefined authorities are allowed to validate transactions and add blocks to the Blockchain. This algorithm entitles a group of nodes to act as validators, granting them the right to change the ledger and safeguard the Blockchain. This results in a sort of agreement concentration in the hands of a few well-known actors. The PoA is especially used in the case of private and consortium Blockchains, since it is based on a set of trusted authorities belonging to one or multiple known entities.

E. InterPlanetary File System (IPFS)

The IPFS [21] is a peer-to-peer distributed file system. The new feature of IPFS is that it now uses content-based addressing instead of location-based addressing. Therefore, we need the hash of some data instead of the address where it is stored to request it. When a file is delivered to IPFS for storage, a unique hash is assigned. As a result, all you have to do to find this file is look up its hash.

IV. System Design and Architecture

In this section, we present the architecture of our proposed system and its different features.

A. Actors of the System

In Fig. 1, we present the proposed system architecture. The system has three types of actors, namely, the patient, the physician, and the hospital. The three types are linked together using a Blockchain network. Health data are collected from the patient, encrypted, and stored in the IPFS which creates a hash of each data. This hash is stored in the Blockchain as a link to the data.

1) Ministry of Health: This entity plays the role of an operational governing which is responsible for implementing the Blockchain solution. It is charged to set the security standards and control access to the Blockchain network. It has the authority to create smart contracts that define the operational policies that all participating entities in the network must adhere to. It also operates the different interactions with the network. It is the only entity that can give permission to the eligible hospitals to participate in the consensus process.

2) Physician or the Medical Team: Physicians are connected to the Blockchain, as a light node, through their computers or smartphones.

[Type here]

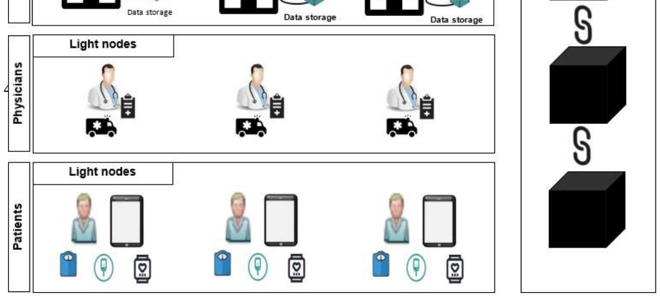


Fig. 1. Architecture design of our system.

3) Hospital: Hospitals are connected with patients through a Blockchain network to have access to their health data. These entities act as full nodes, storing a copy of the Blockchain and contributing to the consensus process. In addition to hospitals, we can also integrate other entities such as pharmaceutical laboratories and public health organizations. These entities are allowed to have a copy of the Blockchain, but they are not allowed to participate in the consensus process. The information gathered from patients will be used for remote monitoring, analysis, and research.

4) Patient: He has two categories of devices.

1) *IoT medical devices:* Each patient has a collection of IoT medical equipment and electronic wearables with embedded sensors. These sensors can collect various vital signs, track physical activity and sleep habits, among other things. The aim of these devices is collecting health data to exchange them with the medical team through the patient's smartphone. Each of these devices is registered in the Blockchain by its owner (patient) and is identified in the network by a pair of unique values (its MAC address and the identification of its owner).
2) *Smartphone:* It serves as a bridge between IoT medical equipment and the medical team. As a result, it is the node that gives access to the Blockchain to other IoT devices. Because of its limitations, the smartphone will not be able to store the entire Blockchain, but will only have access to it through a DApp. When the patient fills out his profile, the DApp creates a Blockchain account for him and generates a pair of keys that will serve as his unique identity.

The patient will use the DApp interface to make different transactions. When the patient creates his profile, the DApp creates a blockchain account and a pair of keys is generated to be a unique identifier for him. Once the patient is registered in the Blockchain, he will be able to execute different actions; add/remove devices, grant/revoke permissions to other nodes, define policies, access his data, and show some summaries and dashboards. The DApp will trigger smart contracts to launch these transactions and store them in the blockchain for traceability. Fig. 2 illustrates the interaction between the patient and the Ethereum Blockchain using a DApp.

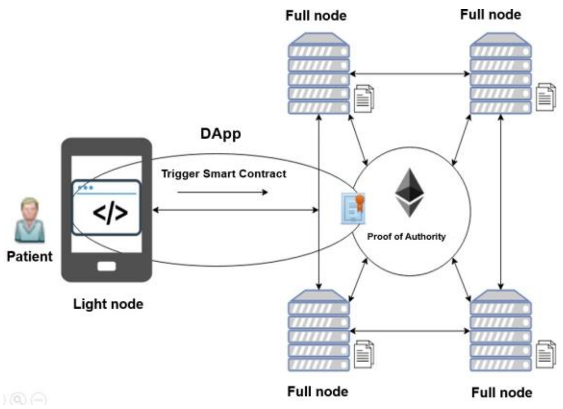


Fig. 2. Patient interaction with blockchain through a decentralized application.

B. Decentralized Storage

To store encrypted health data, we employ IPFS as an off-chain database. We chose IPFS over Blockchain for two reasons. First, the system deals with a vast amount of data. The size of the Blockchain will be affected by storing such a large amount of data, and specific full nodes will be required to hold the data. As a result, the Blockchain is only employed for access control and data integrity protection. On the other hand, even if sensitive data is encrypted, storing it in the Blockchain is impractical. As we all know, data are saved in the Blockchain forever, thus if the encrypted system is ever broken, all nodes will be able to decode and view all data in the system. In addition, the nature of IPFS makes it immune to several attacks such as single point of failure and distributed denial of service (DDoS).

C. Consensus Algorithm

In this work, we adopted the Clique PoA [22] as a consensus algorithm because of the benefits it brings. Clique is a PoA implementation of Ethereum. It is built on the Go language and implemented by the Geth Ethereum client [23]. PoA is one of the most common consensus algorithms for reaching agreements and securing permissioned Blockchain networks. The adoption of such an algorithm has numerous benefits. On the one hand, it can accelerate data storage by speeding up the consensus process. In healthcare applications that require real-time processing, this feature is essential. However, it does not necessitate a large computer resource, making it more energy-efficient. Unlike the PoW algorithm, PoA only enables a limited number of predefined authorities, known as validators, to participate in the consensus process by validating transactions, producing, and appending blocks to the Blockchain. These validators in our situation are hospitals, which are trustworthy authorities belonging to one or more known entities, such as the Ministry of Health.

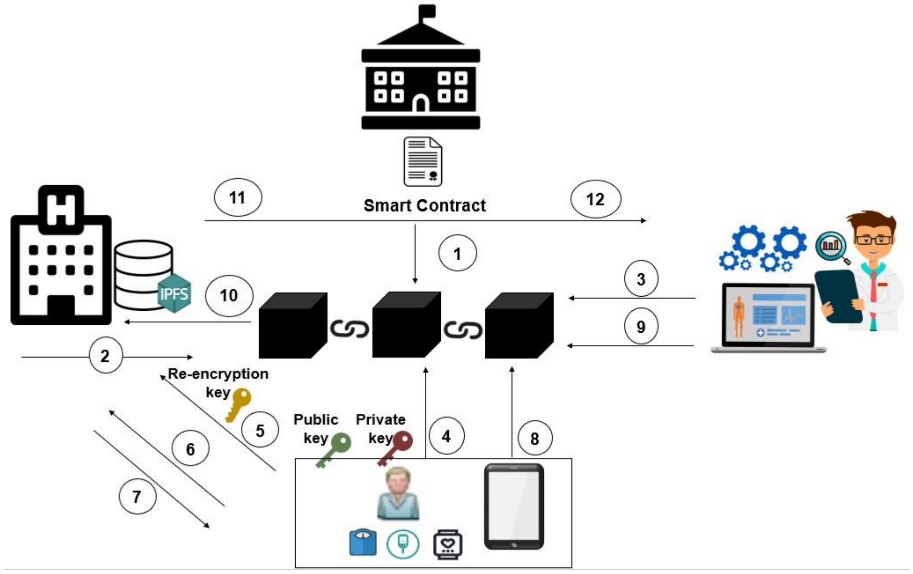


Fig. 3. System working flow.

V. System Working Flow and Use Cases

In this section, we provide the system working flow and the main use case based on diabetes management.

A. Problem Statement and System Working Flow

In our system model, we consider a consortium Blockchain network that groups patients, physicians, and a set of health organizations, belonging to the Ministry of Health, such as clinics and hospitals. The network includes a set of hospitals $H = \{H_1, H_2, \ldots, H_n\}$ and a set of physicians or doctors $D = \{D_1, D_2, \ldots, D_m\}$. Each physician has a set of patients, for example, physician D_i has k patients $P_{D_i} = \{P_{i1}, P_{i2}, \ldots, P_{ik}\}$ Each patient is outfitted with a variety of sensors $S = \{S_1, \ldots, S_l\}$

In Fig. 3, we give a scenario explaining the architecture and the working flow of our solution.

1) *The scenario starts with Step 1:* The Ministry of Health creates the Blockchain network and defines the authorized authorities that can participate in the consensus process. Then, it deploys the smart contract on the Blockchain.
2) *Step 2:* Each hospital registers its physicians in the Blockchain.
3) *Step 3:* Each physician adds his patients to the system.
4) *Step 4:* Once the patient is registered to the network, he can register all his devices in the Blockchain using the DApp installed on his smartphone.
5) *Step 5:* The patient grants access to his physician to have access data. He generates the re-encryption key and sends it to the hospital.
6) *Step 6:* Before sending data, the system authenticates the patient and the device that wants to send data. Data are encrypted and then sent to the hospital for storage in the IPFS.
7) *Step 7:* The hospital sends back the hash of these data to the smartphone.
8) *Step 8:* The data hash is stored on the Blockchain to guarantee data integrity and to control access to it. The smart contract will give access only to eligible entities.
9) *Step 9:* When a physician wants to get data, he sends a request to the smart contract.
10) *Step 10:* After verifying his eligibility, the smart contract sends a confirmation to the hospital.
11) *Step 11:* The hospital retrieves the encrypted data from the IPFS, re-encrypts them, and sends them to the physician.
12) *Step 12:* The physician gets the re-encrypted data and decrypts it with its private key.

B. Case Study: Diabetes Management

Diabetes is a serious chronic disease that has reached alarming proportions. According to the International Diabetes Federation [24], there are 463 million people worldwide who have diabetes in 2019. This figure is expected to increase to 578 million by 2030 and 700 million by 2045. In 2019, there are over one million children and adolescents (aged 18–19) living with type 1 diabetes, and 136 million persons over the age of 65. We will go over three different diabetes-assistance scenarios and how our solution might react in each one (see Table I).

VI. System Implementation

A. Work Environment

The implementation of our solution is based on the following hardware and software components.

1) *Two laptops:* The first is an Intel Core i5-4200U with 16 GB of RAM and 2.30 GHz of CPU frequency, and the second is an Intel Core i5-6300U with 8 GB of RAM and 2.40 GHz of CPU frequency. In these laptops, we have configured a private Ethereum Clique Blockchain. Clique is based on a PoA protocol and it shadows the design of the Ethereum mainnet.

[Type here]

TABLE I
DIABETES MANAGEMENT USE CASES

Use case	Problem description	Solution
Managing diabetes in older adults	Elderly patients with diabetes are the most impacted by diabetes-related complications such as hypoglycemia (low blood sugar) [25], kidney failure and heart disease.	- Track the health situation by collecting health data using wearable devices.
Managing diabetes while traveling	During the travel diabetes management can be disrupted by several factors such as unfamiliar food, delayed meals, being in different time zones and being more active than usual.	- Preserve confidentiality and data integrity.
Assisting diabetic children	Diabetes management can be affected by attending school events or taking field trips.	- Detect and prevent possible hypoglycemia by early recognizing its symptoms.

TABLE II
HEALTH PARAMETERS USED IN OUR WORK

Devices	Parameters
Raspberry pi	Blood glucose
Smartphone	Weight Activities Health assessment

2) A virtualized environment with Ubuntu 18.04 in which a private IPFS network was deployed.
3) A Raspberry Pi 3 Model B was used to simulate an IoT device.

For experimental analysis, we consider some predefined network parameters such as three types of peer nodes representing patients, physicians, and hospitals. These nodes have access to the Blockchain through a DApp that is based on a Web3j API [26] to interact with the Ethereum Blockchain. It also uses the JSON-RPC protocol to connect to the network.

B. Data Collection

In this stage, a Bluetooth connection is established between the smartphone and medical sensors and collects the relevant health data. Many healthcare parameters are required for diabetes treatment, such as glucose levels, cholesterol, blood pressure, body weight, and activities. Some of these parameters will be treated for testing, as stated in Table II. To simulate a glucometer, a Raspberry Pi was used. The patient might manually enter his weight and use his smartphone to post his health assessment. Smartphone sensors can also be used to track activities.

C. Data Storage

After encrypting the data with the patient's public key, our DApp transfers it to IPFS for storage. IPFS then generates a unique hash for each piece of data. The DApp stores the hash in the Blockchain when it receives it. We utilized the java-ipfs-http-client [27], which is a Java client for the IPFS HTTP API, to interact with IPFS nodes. Ubuntu 18.04 nodes are used to run the IPFS network. A Raspberry Pi Model 3B is used to emulate an IoT device. The steps of the data storage process in our technique are depicted in Fig. 4.

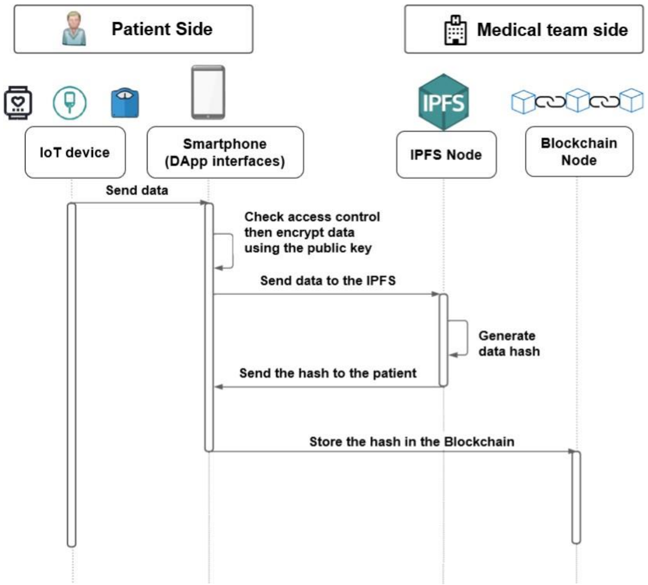

Patient Side
Medical team side
IPFS
IoT device
Smartphone (DApp interfaces)
IPFS Node
Blockchain Node
Send data
Check access control then encrypt data using the public key
Send data to the IPFS
Generate data hash
Send the hash to the patient
Store the hash in the Blockchain

Fig. 4. Sequence diagram for data storage.

D. System Security and Privacy Methods

The proposed solution increases data confidentiality and integrity by providing authentication, access control, and authorization for data sharing and storage management. The security in this solution is ensured through the use of a proxy re-encryption mechanism and Blockchain security features.

1) *Data Encryption/Decryption:* Data encryption/decryption is ensured by using the proxy re-encryption mechanism. Proxy re-encryption is a specific cryptographic scheme that allows a third party (proxy) to transform a ciphertext, which has been encrypted by one key, to become a ciphertext under another key using a re-encryption key. In this mechanism, the proxy performs the re-encryption without having access to the plaintext [28]. In the following, we describe the scheme used in our system. It is mainly composed of five steps:

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

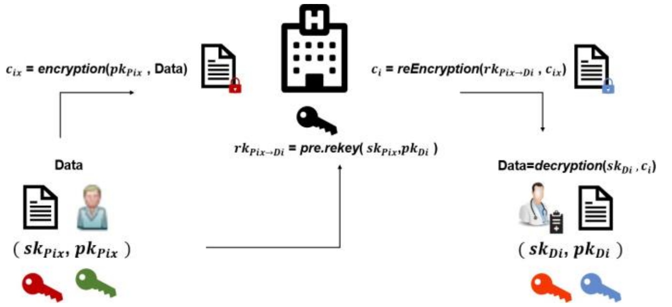
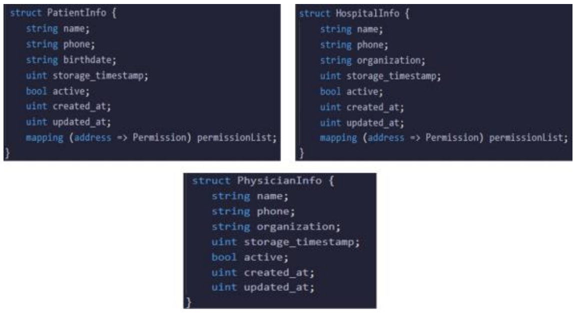

c_{ix} = encryption(pk_{Pix} , Data)
c_i = reEncryption($rk_{Pix \to Di}$, c_{ix})
$rk_{Pix \to Di}$ = pre.rekey(sk_{Pix},pk_{Di})
Data
Data=decryption(sk_{Di} , c_i)
(sk_{Pix}, pk_{Pix})
(sk_{Di}, pk_{Di})

Fig. 5. Proxy re-encryption scenario.

1) *Keys generation:* For a given security parameter $k \in K$, KeyGen function is called to generate a pair of keys (public key $pk_{P_{ix}}$ and private key $sk_{P_{ix}}$) for the patient P_{ix} . We did the same thing to generate the keys for the physician D_i .
2) *Re-encryption key generation:* The ReKey algorithm is used by the patient *ix* to generate a re-encryption key $rk_{P_{ix} \to D_i}$ for his physician D_i. The algorithm uses the key pair of the patient P_{ix} ($pk_{P_{ix}}$, $sk_{P_{ix}}$) and his physician public key pk_{D_i} . Then, the re-encryption key is sent to the hospital which will re-encrypt the data for the physician.
3) *Data encryption:* For a given plaintext (data) message $m \in M$, the patient P_{ix} uses his public key $pk_{P_{ix}}$ to encrypt it. An original ciphertext $c_{ix} \in C_1$ is generated, where C_1 is the set of original ciphertext (encrypted data).
4) *Data re-encryption:* The hospital checks first if the requester has permission to access data. Then, an algorithm is used to re-encrypt the ciphertext $c_{ix} \in C_1$ using the re-encryption key $rk_{P_{ix} \to D_i}$. It is used by the hospital (proxy) to create a transformed ciphertext $c_i \in C_2$ for the physician D_i . C_2 is the set of transformed ciphertext.
5) *Data decryption:* This algorithm is used to decrypt data to have its corresponding plaintext message. For patient P_{ix} , it takes in input the private key $sk_{P_{ix}}$ and the ciphertext c_{ix} . For physician D_i , it takes on input the private key sk_{D_i} and the re-encrypted message c_i .

The encryption and decryption of data are ensured by NuCypher Umbral threshold proxy re-encryption [29]. In this work, we used pyUmbral, which is the Python reference implementation of Umbral. We used Android and Ethereum to create mobile DApps. The integration of pyUmbral with the Android code is accomplished using the Chaquopy [30] python SDK. This SDK allows Python code to be used in Android apps. The Umbral threshold proxy re-encryption is implemented in our approach as shown in Fig. 5.

2) Access Control: In this mechanism, we use smart contracts to efficiently monitor resource access and prevent unauthorized information flow. The contract mainly performs the following four types of algorithms.


```
struct PatientInfo {
    string name;
    string phone;
    string birthdate;
    uint storage_timestamp;
    bool active;
    uint created_at;
    uint updated_at;
    mapping (address => Permission) permissionList;
}
```

```
struct HospitalInfo {
    string name;
    string phone;
    string organization;
    uint storage_timestamp;
    bool active;
    uint created_at;
    uint updated_at;
    mapping (address => Permission) permissionList;
}
```

```
struct PhysicianInfo {
    string name;
    string phone;
    string organization;
    uint storage_timestamp;
    bool active;
    uint created_at;
    uint updated_at;
}
```

Fig. 6. Structs defined in our smart contract.

1) *Register/remove a user:* To authenticate entities that are allowed to access information. Thus, only authorized accesses can take place.
2) *Register/remove a device:* To assist the system in verifying the device identity to prevent access of the unauthorized device. In this way, we can make sure that only data from registered devices are accepted. Patients can then be protected from malicious devices that can, for example, provide incorrect data on their behalf.
3) *Grant/revoke access:* To help patients grant or deny access rights to their data to physicians.
4) *Send/get data:* At this stage, access control lists (ACLs) are created to ensure permission-based security. These lists provide information about access privileges given to some entities.

E. Designed Smart Contract

Using the Solidity programming language, we created a smart contract with several functions to ensure various system functionalities. Each request transmitted to the Blockchain is processed by this smart contract to authenticate the requester and checks his eligibility.

First, we create a set of structs (see Fig. 6) to define new data types in order to encapsulate information about the principal users in our system: patients, physicians, and hospitals.

In the following, we present the pseudo-code of the mainly used algorithms. In this pseudo-code, D_i refers to the Ethereum address of the physician D_i and P_{ix} refers to the Ethereum address of the patient P_{ix}.

To register a patient or update his information, we use Algorithm 1. It needs in parameters: the patient's name, phone number, and the Ethereum address of his associated physician D_i.

When a patient wants to register a new device on the Blockchain, Algorithm 2 is used. In addition to the patient's address, the device's MAC address is passed as an argument. It adds a device to the patient's list who initiated the transaction (the sender). It associates the device's MAC address with its owner (P_{ix}, for example).

Each device is identified in the Blockchain by a unique set of values including its mac address and the address of its owner (patient Ethereum address) (see Fig. 7). In this way, we can ensure that no one can add a device except its owner.

The function for granting permission to a physician is represented in Algorithm 3. The function accepts parameters

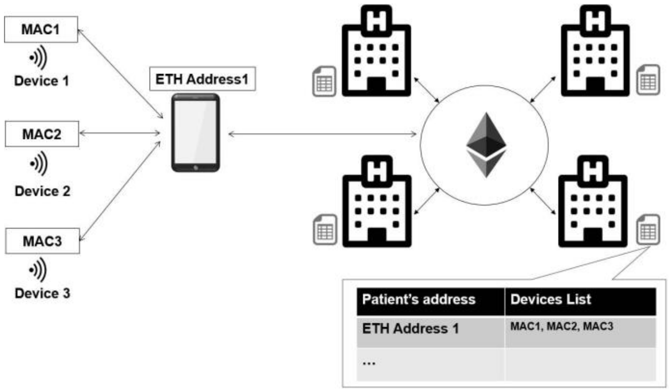
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 Register or Update Patient

```
procedure ADDPATIENT($P_{ix}$, $D_i$, name, phone, birth)
    // Assign a physician to the patient
    $listPhysicians[P_{ix}] \Leftarrow D_i$;
    $createdAt \Leftarrow now$;
    if Patient doesn't exist then
        Add $P_{ix}$ to the list of patient addresses
    else
        // Preserve createdAt timestamp
        $createdAt \Leftarrow listPatients[P_{ix}].creationDate$
    end if
    // Add or update the patient information
    $patient \Leftarrow PatientInfo(\{$
        $name \Leftarrow name,$
        $phone \Leftarrow phone,$
        $birthdate \Leftarrow birth,$
        $storage\_timestamp \Leftarrow now,$
        $active \Leftarrow true,$
        $creationDate \Leftarrow createdAt,$
        $updatedAt \Leftarrow now$
        $\})$
    $listPatients[P_{ix}] \Leftarrow patient$
end procedure
```

Algorithm 2 Register a New Device

```
procedure ADDDEVICE($P_{ix}$, macAddress)
    // Check either the device exist or not
    $require(!deviceExists[macAddress]);$
    // Append the Mac address to the devices list of $P_{ix}$ if the device doesn't already exist
    $devicesList[P_{ix}].push(macAddress)$;
    $deviceExists[macAddress] = true$;
end procedure
```



Patient's address	Devices List
ETH Address 1	MAC1, MAC2, MAC3
...	

Fig. 7. Device identifier.

such as the physician address for whom authorization is being granted, read and write permissions, and hash information (which will be used in the re-encryption process). For each patient, a list of entities to which he accorded the access is created.

Algorithm 3 Add Permission

```
procedure ADDPERMISSION($P_{ix}$, $D_i$, read, write, hashInfo)
    $perm \Leftarrow Permission(D_i, read, write, hashInfo)$
    $permissionList[P_{ix}].push(perm)$
end procedure
```

Algorithm 4 Send Data Hash

```
procedure ADDDATA($P_{ix}$, device, dataHash, capsule)
    for ($i = 0$; $i < devicesList[P_{ix}].length$; $i{+}{+}$) do
        if device exists in the patient list then
            $record \Leftarrow EMR(device, dataHash, now, capsule)$ $EMRs[P_{ix}].push(record)$
        end if
    end for
end procedure
```

Algorithm 5 Get Data Hash

```
procedure GETDATA($P_{ix}$, data)
    if $P_{ix} != requester$ then
        if $P_{ix}.permissionList[requester].read == True$ then
            return data, reKey
        else
            access denied
        end if
    else
        return data
    end if
end procedure
```

The function, which is responsible for storing the data hash on the Blockchain, is described by Algorithm 4. It takes as parameters, the data owner (P_{ix}), the data hash, the device generating this data, and a capsule (which will be used for decryption).

Algorithm 5 represents access to the data mechanism. It takes as parameters, the data owner (P_{ix}) and the data to get. If the requester has access, the function returns the data hash and the re-encryption key, otherwise the access will be denied.

F. Experimental Results

In this section, we show screenshots of the Android DApp's patient, as well as the results of executing several functions with Remix integrated development environment (IDE) [31]. Remix is an IDE that provides the development, compilation, and deployment of smart contracts. It also serves as a platform for executing transactions and interacting with smart contracts. Initially, we create a private Ethereum Blockchain with five nodes, including three validators. Then, we used Remix to create and deploy the smart contract. The interaction with the smart contract is done through the android DApp. We also use the Web3 Provider to connect the Remix with one of the Blockchain nodes. Fig. 8 illustrates the system used for implementation. We used two PCs: a Raspberry PI and a

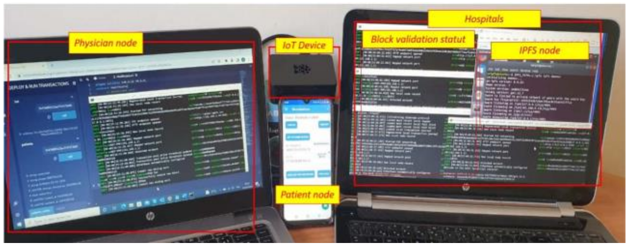


Fig. 8. System implementation on PC, Raspberry, and Smartphone.

smartphone. The first PC represents the physician node and the second one illustrates the hospital node. The smartphone simulates the patient node and the Raspberry is for emulating an IoT device.

Fig. 9 shows the different interfaces of our DApp. In addition, to verify the authentication, the smart contract is also the engine that controls our DApp back-end, providing different functionalities.

1) Once a patient installs our DApp, he needs to scan a QR code generated by his physician to access the network. The QR code contains the physician's address and some other information about the network such as the Blockchain client URL, Blockchain ID, and IPFS HTTP client address.
2) After scanning the QR code, the patient creates his identity by creating an Ethereum account and generates his pair of keys.
3) It depicts the various features of our program. The initial step is to scan new devices and display a list of those that have been paired. The patient's address and QR code are displayed in the Encryption item. The profile item is used to keep track of patient data. The send data item is used to send data to the physician (weight, activities, health evaluation), whereas the Get Data item is used to get patient data.
4) Every patient must first register all of his or her medical equipment that is utilized to collect data. Fig. 9(d) represents the list of paired devices with Bluetooth. Once the patient registers his device, addDevice function adds this device to the list of this patient's devices. As a result, no one but the patient can add a gadget to the patient's registry.

Using Remix IDE, Fig. 10 shows a screenshot of information of a registered patient. The patient address is "0x47b682522aa7747473b90816453df87a94304eb4" and he is associated with the physician who have the address "0x1a5e5a6D31e3308816b015E29679fa0949965519."

Fig. 11 shows the first device registered by bob. The device's MAC address is 10:8E:E0:A3:B7:1F and it is linked with the address of its owner (bob).

VII. Security Analysis and Discussion

In this section, we will go through how the proposed method assures scalability and quick processing time. We also describe the main security goals that our technique achieves and compare them to existing solutions.

TABLE III
Processing Times in Our Model

Type of operation	Processing time (ms)
Block creation	5000
Physician registration	10482
Patient registration	10648
Device registration	9625
Send data	19747
Sends a fake sensor data	9532

A. Scalability and Processing Time

The consensus algorithm and the chosen data storage type have a positive impact on processing speed and scalability. The scalability of a consensus algorithm refers to the transaction throughput that it offers. We can achieve high throughput by using the PoA algorithm because it is based on a known and a limited number of validators. PoA can also ensure a short processing time. The transaction time in a PoA network is faster than the transaction time in a PoW network. The block creation time is set to 5 s, and the gas limit is set to 47 00 000 in our case. Using this configuration, we can process 45 transactions per second (TPS). Table III illustrates the processing times for some operations. In addition, PoA is more suitable for permissioned Blockchains [32] and can increase the performance in terms of the time of process validation as well as the number of validated transactions per second. Data scalability is also maintained by storing data hashes rather than the data itself in the Blockchain. Furthermore, the combination of PoA and distributed storage using IPFS allows the system to be expanded in real-time.

B. Security and Privacy Analysis

In our proposed system, three principal techniques are used to provide security: Blockchain, proxy re-encryption, and decentralized access control. The combination of these technologies improves security, prevents data breaches, and protects patient privacy. In addition, the system is fully decentralized and distributed which provides security from a single point of failure.

1) Security Model Insurance:

1) *Data security and availability:* Due to their height level of security, the combination of Blockchain and proxy re-encryption is a robust solution to increase the security of our system. The decentralization and replication of data are one of the most essential properties of Blockchain. This eliminates the risk of a single point of failure and ensures the availability of data. The adoption of various medical devices, smartphones, and DApps to ensure the interaction in the system can also provide the availability of our solution.
2) *Decentralized access control based on smart contracts:* The access control list is provided and managed using a smart contract. This process starts with identity verification, which is done in two steps: authentication and authorization. The first step is to authenticate the requester's identity and the second one is to specify access rights and privileges. Once the requester's identity

[Type here]

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

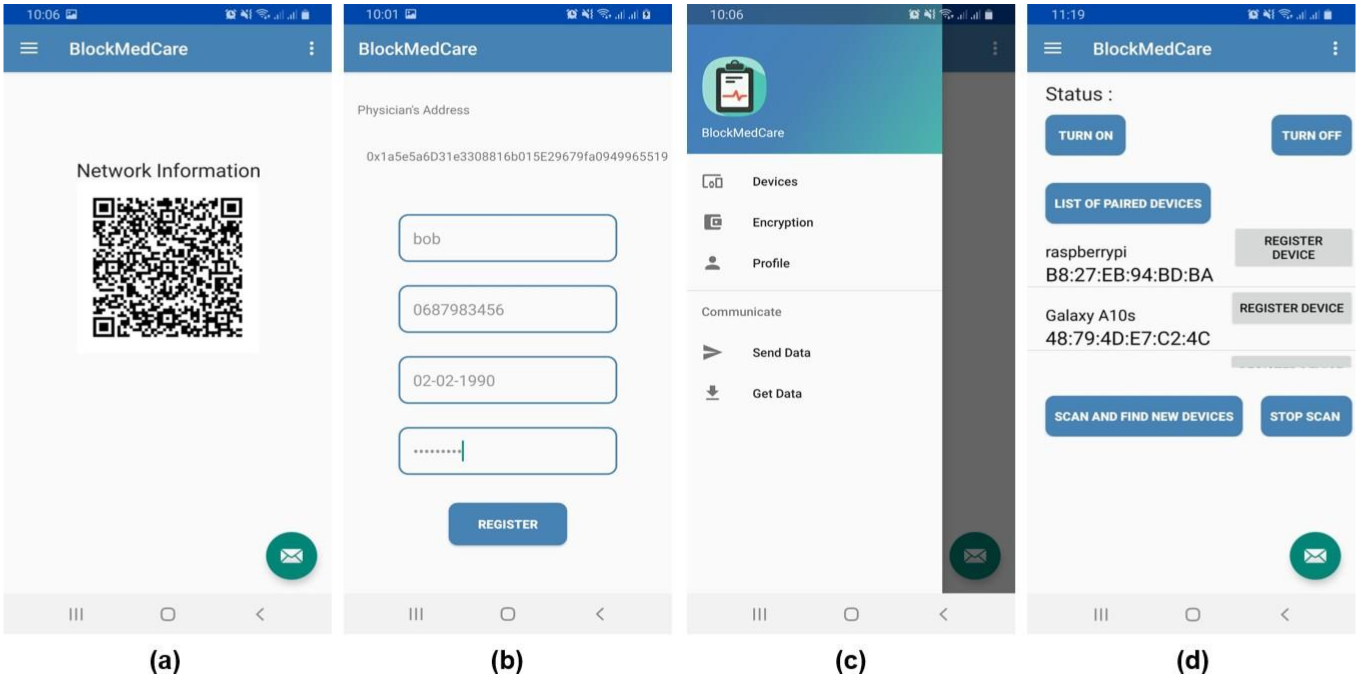
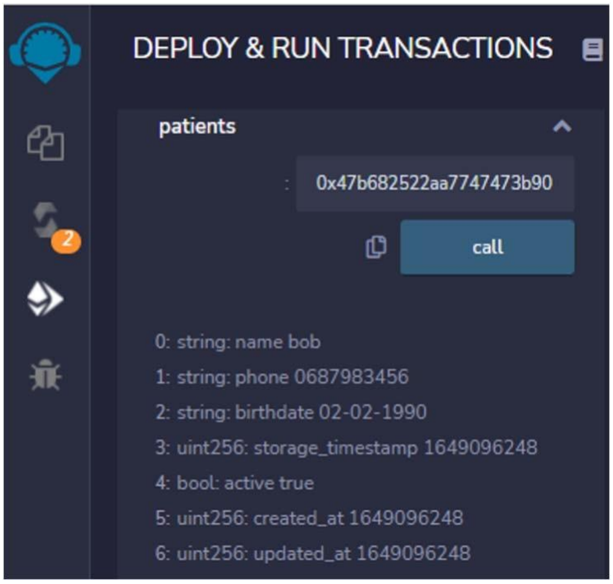


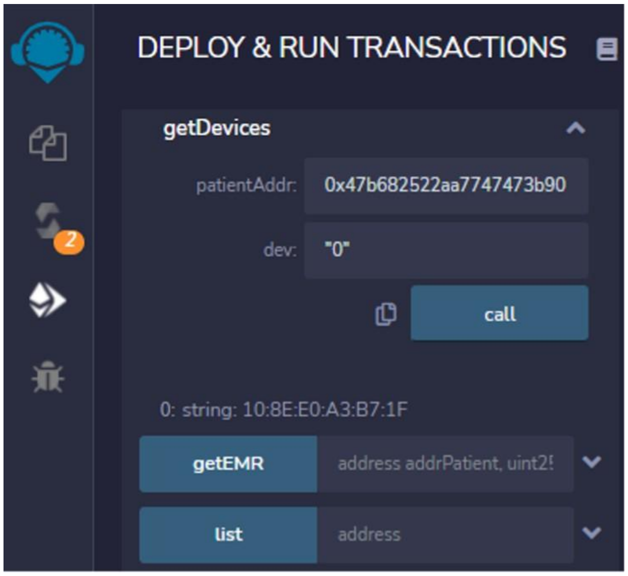
Fig. 9. Different user interfaces in the patient's DApp. (a) QR code generated by the physician. (b) Patient registration interface. (c) Various features of our application. (d) Device registration interface.



```
: 0x47b682522aa7747473b90

0: string: name bob
1: string: phone 0687983456
2: string: birthdate 02-02-1990
3: uint256: storage_timestamp 1649096248
4: bool: active true
5: uint256: created_at 1649096248
6: uint256: updated_at 1649096248
```

Fig. 10. Read patient information using Remix.



```
patientAddr: 0x47b682522aa7747473b90
dev: "0"

0: string: 10:8E:E0:A3:B7:1F

getEMR   address addrPatient, uint2!
list     address
```

Fig. 11. Read device information using Remix.

is verified, he will be granted access to data. In contrast to the traditional access control solutions that are based on centralized systems, the decentralized access control can prevent several security issues such as privacy leakage and single point of failure.

3) *Data integrity:* Data integrity is ensured by storing data hash on the Blockchain. Due to the immutability feature, which protects the data from being tampered with, this hash could not be modified after being registered. In addition, the use of smart contracts for controlling access to data is also enhancing integrity. As a result, only authorized people to have access to data. Blockchain also enables traceability by recording all data alterations and actions.

4) *Data confidentiality:* In our system, the confidentiality of the patient's data is protected by the use of smart contracts that govern access to the data. As a result, medical information is secure and cannot be accessed by unauthorized individuals.

5) *Patient's privacy:* Different concepts are used to protect the patient's privacy: The first is to take advantage of Blockchain's anonymity feature. As a result, each

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
COMPARISON OF OUR SYSTEM WITH THE EXISTING STATE-OF-THE-ART WORKS

Research work	Blockchain	Consensus	IoT	Data storage	Data Encryption	Confidentiality	Integrity	Privacy	Access Control	Implementation
[33]	Ethereum (public)	PoW	✗	Central database	✗	✓	✗	✗	✗	✓
[34]	Bitcoin (public)	PoW	✓	Blockchain	✗	✓	✓	✓	✓	✓
[35]	Hyperledger (private)	PBFT	✓	Central database	✓	✓	✓	✓	✓	✗
[36]	Ethereum	Not specified	✓	Central database	✗	✓	✓	✗	✓	✗
[37]	Ethereum (private)	Not specified	✓	IPFS	✓	✗	✓	✓	✓	✓
[38]	Quorum (private)	QuorumChain	✗	Central database	✓	✓	✗	✓	✓	✓
[39]	Ethereum (private)	Not specified	✗	Central database	✗	✗	✗	✓	✓	✗
[40]	Ethereum (private)	Not specified	✗	Central database	✓	✗	✓	✗	✓	✓
[41]	Not specified	Not specified	✓	Central database	✓	✓	✓	✓	✓	✗
[42]	Ethereum (private)	POA	✓	Central database	✓	✓	✓	✓	✗	✗
[43]	Hyperledger(private)	PBFT	✓	Central database	✓	✓	✓	✓	✓	✓
[44]	Ethereum (private)	Not specified	✓	IPFS	✗	✓	✗	✓	✓	✗
[45]	Not specified	Not specified	✗	IPFS	✓	✗	✓	✓	✓	✓
[46]	Not specified (private)	PoW	✗	IPFS	✗	✗	✓	✓	✗	✓
[47]	Not specified	Not specified	✓	IPFS	✓	✗	✗	✓	✓	✗
[48]	Ethereum (private)	Not specified	✓	IPFS	✓	✗	✗	✓	✗	✗
[49]	Not specified	Not specified	✗	IPFS	✓	✗	✓	✓	✓	✓
Our work	Ethereum (private)	PoA	✓	IPFS	✓	✓	✓	✓	✓	✓

patient is identified only by a unique identity, making his identification by other nodes impossible and exclusive to his physician. The second is data encryption, which encrypts data and stores it in IPFS, while only his hash is stored in the Blockchain. The third is limiting who has access to this information.

2) Resistance to Attacks:

1) *Distributed denial of service (DDoS) attacks:* A DDoS attack disrupts a service's usual traffic by flooding it with requests until it can no longer receive further requests. Using Blockchain and IPFS, the proposed solution is entirely based on a decentralized and distributed architecture, which can prevent DDoS attacks.
2) *Impersonation attacks:* This attack occurs when an adversary impersonates the identity of a trusted entity. Due to the complexity of solving the ECDSA's elliptic curve discrete logarithm problem, this attack is difficult to be performed.
3) *Message forgery attack:* It involves forging or altering a message. Because each transaction in the network is signed and authenticated before being inserted into the Blockchain, this attack is also impossible.
4) *Man-in-the-middle attacks:* These attacks involve intercepting communication between two entities without any doubt that the communication has been compromised. Our method encrypts and signs the transaction with a private key that is only known by the owner. The transaction must be signed with a valid signature belonging to the sender's address in order to be validated. As a result, forging a signature without knowing the corresponding private key is extremely difficult.

VIII. RELATED WORK

In this section, we present an overview of the existing works processing privacy and access control in healthcare systems. For security considerations, various studies have proposed systems and methodologies that combine IoT and Blockchain technologies in the healthcare sector.

Azaria *et al.* [33] proposed to manage permissions and control access to medical records using smart contracts. They used Ethereum Blockchain network to build and deploy smart contracts that help patients to govern and manage access to their data. They also proposed to use a Gatekeeper database to store data. Ouaddah *et al.* [34] proposed a framework based on Blockchain for access control in IoT. It enables the resource owner to control his data by creating an access token using a smart contract. Ali *et al.* [35] presented a decentralized approach for permission delegation and access management for IoT applications, but they are not discussed the system implementation. Using Blockchain and trusted oracles and smart contracts, decentralized access control for IoT data was proposed by Al Breiki *et al.* [36]. Nguyen *et al.* [37] presented a secure Blockchain for sharing electronic health records on a mobile cloud. They also suggested a trustworthy access control mechanism using smart contracts. In [38], the authors proposed a framework, called Ancile, for access control management and for promoting interoperability. This framework uses a permissioned Ethereum-based Blockchain called Quorum. Another work by Maslove *et al.* [39] described BlockTrial, a solution based on Blockchain, to manage clinical trial data using web-based interfaces. Zhuang *et al.* [40] proposed a Blockchain-based platform for exchanging health information and monitoring clinical trial processes. They developed a web-based graphical user interface. In [41], the authors proposed a Blockchain-based access control architecture for e-health applications. The authors in [42] developed a Blockchain-based system for maintaining privacy in IoT healthcare devices. To encrypt data and validate transactions, a lightweight cryptographic algorithm is used. Another project is looking at combining IoT and Blockchain to develop a medical platform that will assure the integrity of electronic medical records [43]. Rifi *et al.* [44] proposed to use smart contracts to govern medical data access based on Ethereum Blockchain. Medical sensors generate data, which is handled by a gateway and stored in an off-chain database based on IPFS. In [45], the authors proposed a solution based on Blockchain and IPFS, as well as attribute-based encryption for storing and securing EMR. Most recently, Jayabalan and Jeyanthi [49] proposed a Blockchain-based framework integrated with IPFS for electronic health record management. The assessment approaches presented have valuable insights, but still have some issues in terms of security and implementation.

Table IV gives a comparison of our work with the most notable existing solutions that use Blockchain in the healthcare sector for securely sharing health information. The key metrics comparison are, especially, related to the technologies used and the security level provided by each work. The comparison starts with information about the Blockchain platform exploited in each solution and the consensus mechanism used. Then, it gives information about the storage system exploited in each work and whether it considers the integration of IoT devices in its solution or not. It also specifies the level of security each architecture offers and whether or not it encrypts data before storing it. Only three works (see [33], [34], and [46]) that use the PoW consensus algorithm. The biggest issue with this mechanism is that the validation process consumes a lot of energy. In addition, it might also be considered slow when it comes to transaction speed. In terms of data storage, many works used a centralized database to store data (see [33], [35], [36], [38], [39]–[42], and [43]). This kind of storage is vulnerable to several security issues such as single point of failure, DDoS attacks, and to be tampered. Moreover, the comparison shows that our approach considers all important security requirements, namely, confidentiality, integrity, privacy, and access control, while other works focus on just a few of these requirements. Furthermore, the table displays some works that are still not yet implemented.

IX. Conclusion and Future Work

In this article, we present a secure healthcare system for remote patient monitoring. Our solution is completely decentralized and based on the use of several emerging technologies, namely, the Internet of Things, Blockchain, smart contracts, and IPFS. To ensure security and privacy, the proposed system adopts Blockchain and proxy re-encryption. To solve the problem of scalability, we propose to store only some condensed information about the data and its hash in the Blockchain. Health data are encrypted before being stored in an off-chain database that relies on a private IPFS. Using a scenario-based diabetes management system, we empirically validate a secure approach that provides remote monitoring for diabetic patients. The proposed system offers a high security level compared to the state-of-the-art methods. Currently, we are developing a web version of our solution. In future work, we plan to add a fog layer between medical devices and the hospital in order to filter data before sending them to the hospital. It can also process data and provide the patient with real-time analysis.

References

[1] *Physicians (Per 1,000 People)—Lower Middle Income | Data*. Accessed: Mar. 19, 2022. [Online]. Available: https://data.worldbank.org/indicator/SH.MED.PHYS.ZS

[2] A. Misra *et al.*, "Diabetes in developing countries," *J. Diabetes*, vol. 11, no. 7, pp. 522–539, Jul. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/1753-0407.12913

[3] World Health Organization. *OMS | Maladies chroniques*. Accessed: Apr. 11, 2021. [Online]. Available: https://www.who.int/topics/chronic_diseases/fr/

[4] *Facts & Figures*. Accessed: Mar. 19, 2022. [Online]. Available: https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html

[5] J. J. P. C. Rodrigues, S. S. Compte, and I. de la Torra Diez, "Introduction," in *e-Health Systems*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 15–34. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9781785480911500166

[6] M. A. Al-Shaher and N. J. Al-Khafaji, "E-healthcare system to monitor vital signs," in *Proc. 9th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2017, p. 5.

[7] E. Andrès *et al.*, "e-health: A promising solution for optimizing management of chronic diseases. Example of the national e-health project e-care based on an e-platform in the context of chronic heart failure," *Eur. Res. Telemed.*, vol. 4, no. 3, pp. 87–94, 2015.

[8] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[9] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Bus. Rev., 2008, p. 21260.

[11] C. A. Shoniregun, K. Dube, and F. Mtenzi, "Laws and standards for secure e-Healthcare information," in *Electronic Healthcare Information Security* (Advances in Information Security), vol. 53, S. Jajodia, Ed. Boston, MA, USA: Springer, 2010, pp. 59–100. [Online]. Available: http://link.springer.com/10.1007/978-0-387-84919-5_3

[12] U.S. Department of Health & Human Services Office for Civil Rights. *Privacy, Security, and Electronic Health Records*. Accessed: Jul. 21, 2021. [Online]. Available: https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/understanding/consumers/privacy-security-electronic-records.pdf

[13] Office for Civil Rights (OCR). (Oct. 2015). *187-What Does the HIPAA Privacy Rule Do*. [Online]. Available: https://www.hhs.gov/hipaa/for-individuals/faq/187/what-does-the-hipaa-privacy-rule-do/index.html

[14] (May 2008). *The HIPAA Privacy Rule*. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

[15] (Sep. 2009). *The Security Rule*. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/security/index.html

[16] M. A. Sahi, H. Abbas, A. Derhab, and M. Orgun, "Privacy preservation in e-Healthcare environments: State of the art and future directions," *IEEE Access*, vol. 6, pp. 464–478, 2017.

[17] S. P. Amaraweera and M. N. Halgamuge, "Internet of Things in the healthcare sector: Overview of security and privacy issues," in *Security, Privacy and Trust in the IoT Environment*. Cham, Switzerland: Springer, 2019, pp. 153–179, doi: 10.1007/978-3-030-18075-1_8.

[18] J. S. Kumar and D. R. Patel, "A survey on Internet of Things: Security and privacy issues," *Int. J. Comput. Appl.*, vol. 90, no. 11, pp. 20–26, Mar. 2014. [Online]. Available: http://research.ijcaonline.org/volume90/number11/pxc3894454.pdf

[19] (2019). *Ethereum*. Accessed: Dec. 10, 2019. [Online]. Available: https://github.com/ethereum/wiki

[20] A. M. Antonopoulos, M. Andreas, and G. Wood, *Mastering Ethereum: Building Smart Contracts and Dapps*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[21] Protocol Labs. (2019). *IPFS Powers the Distributed Web*. Accessed: Dec. 20, 2019. [Online]. Available: https://ipfs.io/

[22] Ethereum. (2019). *Private Networks | Go Ethereum*. Accessed: Dec. 19, 2019. [Online]. Available: https://geth.ethereum.org/docs/interface/private-network

[23] Ethereum. (2019). *Go Ethereum*. Accessed: Dec. 19, 2019. [Online]. Available: https://geth.ethereum.org/

[24] International Diabetes Federation. (2019). *IDF DIABETES ATLAS Ninth Edition 2019*. [Online]. Available: https://www.diabetesatlas.org/en/resources/

[25] P. Piatkiewicz, "Hypoglycemia in elderly type 2 diabetes patients," *Diabetes Manage.*, vol. 6, no. 3, p. 5, 2016.

[26] (2019). *Web3j*. Accessed: Dec. 20, 2019. [Online]. Available: https://github.com/web3j/web3j

[27] (Dec. 2019). *International Community*. Accessed: Dec. 21, 2019. https://github.com/ipfs/java-ipfs-http-client

[28] D. Nuñez, I. Agudo, and J. Lopez, "Proxy re-encryption: Analysis of constructions and its application to secure access delegation," *J. Netw. Comput. Appl.*, vol. 87, pp. 193–209, Jun. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804517301078

[29] D. Nunez, "UMBRAL: A threshold proxy re-encryption scheme," NuCypher, NICS Lab., Univ. Malaga, Málaga, Spain, Tech. Rep., 2018, p. 8. [Online]. Available: https://raw.githubusercontent.com/nucypher/umbral-doc/master/umbral-doc.pdf

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[30] Chaquo. (2019). *The Easiest Way to Use Python in Your Android App*. Accessed: Dec. 21, 2019. [Online]. Available: https://chaquo.com/chaquopy

[31] Remix. (2020). *Remix—Ethereum IDE*. Accessed: Jun. 14, 2020. [Online]. Available: https://remix.ethereum.org

[32] K. Azbeg, O. Ouchetto, S. Jai Andaloussi, and L. Fetjah, "An overview of blockchain consensus algorithms: Comparison, challenges and future directions," in *Advances on Smart and Soft Computing* (Advances in Intelligent Systems and Computing), vol. 1188, F. Saeed, T. Al-Hadhrami, F. Mohammed, and E. Mohammed, Eds. Singapore: Springer, 2021, pp. 357–369. [Online]. Available: http://link.springer.com/10.1007/978-981-15-6048-4_31

[33] 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. [Online]. Available: http://ieeexplore.ieee.org/document/7573685/

[34] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "FairAccess: A new blockchain-based access control framework for the Internet of Things: FairAccess: A new access control framework for IoT," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5943–5964, Dec. 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/sec.1748

[35] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, and Q. E. Ali, "Blockchain based permission delegation and access control in Internet of Things (BACI)," *Comput. Secur.*, vol. 86, pp. 318–334, Sep. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404819301208

[36] H. Al Breiki, L. A. Qassem, K. Salah, M. H. U. Rehman, and D. Sevtinovic, "Decentralized access control for IoT data using blockchain and trusted oracles," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Orlando, FL, USA, Nov. 2019, pp. 248–257. [Online]. Available: https://ieeexplore.ieee.org/document/9065150/

[37] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based e-Health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8717579/

[38] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210670717310685

[39] D. M. Maslove, J. Klein, K. Brohman, and P. Martin, "Using blockchain technology to manage clinical trials data: A proof-of-concept study," *JMIR Med. Informat.*, vol. 6, no. 4, Dec. 2018, Art. no. e11949. [Online]. Available: http://medinform.jmir.org/2018/4/e11949/

[40] Y. Zhuang, L. Sheets, Z. Shae, J. J. P. Tsai, and C.-R. Shyu, "Applying blockchain technology for health information exchange and persistent monitoring for clinical trials," in *Proc. AMIA Annu. Symp.* Bethesda, MD, USA: American Medical Informatics Association, 2018, p. 1167.

[41] K. M. Hossein, M. E. Esmaeili, T. Dargahi, and A. Khonsari, "Blockchain-based privacy-preserving healthcare architecture," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, Edmonton, AB, Canada, May 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8861857/

[42] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, p. 326, Jan. 2019. [Online]. Available: http://www.mdpi.com/1424-8220/19/2/326

[43] L. Hang, E. Choi, and D.-H. Kim, "A novel EMR integrity management based on a medical blockchain platform in hospital," *Electronics*, vol. 8, no. 4, p. 467, Apr. 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/4/467

[44] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for eHealth data access management," in *Proc. 4th Int. Conf. Adv. Biomed. Eng. (ICABME)*, Beirut, Lebanon, Oct. 2017, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/8167555/

[45] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9045940/

[46] R. Kumar, N. Marchang, and R. Tripathi, "Distributed off-chain storage of patient diagnostic reports in healthcare system using IPFS and blockchain," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, Jan. 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9027313/

[47] B. Alamri, I. T. Javed, and T. Margaria, "A GDPR-compliant framework for IoT-based personal health records using blockchain," in *Proc. 11th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Paris, France, Apr. 2021, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9432661/

[48] K. Miyachi and T. K. Mackey, "HOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102535. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306457321000431

[49] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy," *J. Parallel Distrib. Comput.*, vol. 164, pp. 152–167, Jun. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0743731522000648

Kebira Azbeg received the master's degree in computer science engineering and internet from the University of Hassan II of Casablanca, Casablanca, Morocco, in 2016, where she is currently pursuing the Ph.D. degree in computer science with the Faculty of Sciences Ain Chock.

She is also a member with the Computer Science, Modeling Systems and Decision Support Laboratory. Her research interests include blockchain, the Internet of Things, cryptography, and artificial intelligence.

Ouail Ouchetto received the master's degree in applied mathematics from Pierre et Marie Curie University, Paris, France, the master's degree in informatics from the Télécom Bretagne School of Engineers, Brest, France, in 2003, and the Ph.D. degree in modeling and engineering science from the University of Paris-Saclay, Bures-sur-Yvette, France, in 2006.

He was a Post-Doctoral Researcher with the University Blaise Pascal, Clermont-Ferrond, France, and an Assistant Professor (Ater) with the University of Paris-Saclay. In September 2010, he joined Aïn Chock Hassan II University, Casablanca, Morocco, as a Professor of mathematics and computer science. His research interests include electrical and electronic engineering, applied mathematics, and computer science.

Said Jai Andaloussi received the Ph.D. degree in computer science jointly supervised by the Faculty of Science, University of Sidi Mohamed Ben-Abedillah, Fez, Morocco, and Telecom Bretagne, Brest, France, in 2010.

He is currently a Professor of computer science with the Mathematics and Computer Science Department, Faculty of Sciences Ain-chok, Casablanca, Morocco. He is the author of more than 40 articles. His current research interests include medical information processing, information modeling, and analysis of medical images.