Egyptian Informatics Journal 23 (2022) 329–343

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security

Kebira Azbeg [*], Ouail Ouchetto, Said Jai Andaloussi

Computer Science and Systems Laboratory, Department of Mathematics and Computer Sciences, Morocco
Faculty of Sciences Ain Chock, Hassan II University of Casablanca, Morocco

ARTICLE INFO

Article history:
Received 1 October 2021
Accepted 11 February 2022
Available online 23 February 2022

Keywords:
Healthcare
Remote patient monitoring
Internet of things
Blockchain
Security
Proxy re-encryption
IPFS

ABSTRACT

Nowadays, healthcare is growing rapidly due to the large development of new technologies such as IoT and wearable devices. These devices are widely used to ensure remote patient monitoring. The current implementation is based on a client/server architecture. This raises several challenges regarding security and privacy that make healthcare systems more susceptible to several attacks. Therefore, health data are subject to strict regulatory and security requirements. To overcome these challenges and comply with security regulations, the adoption of a distributed architecture is a necessity. Due to its distributed nature and its security promises, Blockchain has a large interest as a sophisticated technology to solve the security challenges in IoT-based systems. Motivated by these factors, this work proposes BlockMedCare, a secure healthcare system that integrates IoT with Blockchain. The system is designed to support remote patient monitoring, especially when it comes to chronic diseases that require regular monitoring. We took into consideration three main parameters: security, scalability, and processing time. The security is ensured by using the re-encryption proxy combined with Blockchain to store hash data. Smart contracts are used for access control. To ensure Blockchain scalability, an off-chain database based on IPFS is used to store data. To speed up the data storage process, we use an Ethereum Blockchain-based proof of authority. As a use case, we applied the system to diabetes management and showed the execution results based on the system interfaces. The experimental system has demonstrated a good improvement of healthcare systems in terms of security face to the existing methods.

1. Introduction

According to the World Health Organization (WHO), chronic diseases are the leading causes of death and disability in the world. The number of deaths caused by chronic diseases was expected to rise to 73% of all deaths and 60% of the global burden of disease in 2020 [1]. The four major chronic diseases are: cardiovascular disease (CVD), cancer, chronic obstructive pulmonary disease and type 2 diabetes. CVD, for example, caused 17.5 million deaths in 2012 (mainly due to heart attacks and strokes). By 2030, this number is expected to increase to reach almost 22.2 million [2]. Meanwhile, diabetes caused 1.5 million deaths in 2012, according to the global report on diabetes [3].

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 healthy and have

[*] Corresponding author.
E-mail addresses: kebira.azbeg-etu@etu.univh2c.ma (K. Azbeg), ouail.ouchetto@etude.univcasa.ma (O. Ouchetto), said.jaiandaloussi@etude.univcasa.ma (S. Jai Andaloussi).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.02.004

a near-normal life if the disease is early detected and well managed. This kind of disease requires regular check-ups and serious self-care. Thus, it can be monitored and we can then prevent its progress. Nowadays, technology advancement such as Internet of things (IoT), smart medical devices and mobile applications can support daily healthcare activities to control chronic diseases and provide remote patient monitoring. In addition, the use of smartphones has known wide adoption among healthcare professionals. According to a UK-based survey study, 92.6% of physicians and 53.2% of nurses saw their smartphones as ''very useful'' or ''useful'' in helping them perform their daily clinical duties [4].

IoT refers, in general, to a system of billions interrelated devices around the world that are connected to the internet and are able to collect, store, and exchange data. These devices could be smartphones, vehicles, wearables, and any other electronic device with embedded processors, network connectivity, and sensors. In healthcare, IoT devices, especially medical devices and electronic wearables, are widely used. A wide variety of applications are based on the use of IoT, such as the monitoring of chronic patient healthcare, elderly care, and emergency cases [5]. The power of IoT medical devices is manifested in data collection, monitoring, and analysis. These devices are equipped with sensors that can track physical activity and measure different vital signs such as blood pressure, glucose levels, heart rate, body temperature, weight, and even sleep patterns. There are a wide variety of reliable and wearable devices for healthcare monitoring on the market [6].

The current IoT system architecture is based on client–server communication. IoT devices are connected to a central cloud server which is used to ensure the communication between devices, handle and store data. This centralized architecture may create a single point of failure. That may increase security and privacy risks. Therefore, the adoption of new solutions based on a decentralized architecture is now a necessity.

In this paper, we propose a secure healthcare system, named BlockMedCare, based on the combination of IoT medical devices and Blockchain technology. Our solution is designed to collect and share patient data with medical teams. Provides a high level of security when sharing and storing this sensitive data. The system architecture is based on a set of technologies, namely: IoT, Blockchain, Smart contracts, IPFS and proxy re-encryption. Our solution is fully decentralized, efficient, and feasible in practice, and it meets the expected security requirements. Furthermore, our approach considers the most important security criteria, such as confidentiality, integrity, privacy, and access control, while other works are concentrating just on some of these requirements. Furthermore, most of the cited works that use IPFS for storage have yet to be deployed.

The rest of this paper is organized as follows. Section 2 presents preliminary definitions concerning IoT-Healthcare systems, Blockchain technology, and IPFS. Section 3 describes the proposed approach. Section 4 illustrates the use cases and a practical scenario of this approach. Section 5 gives details about the implementation of the system. Section 6 discusses the security of our system. Section 7 presents the related work. Then, in Section 8, we conclude the paper and introduce some future work.

2. Preliminary definitions

2.1. IoT-healthcare security requirements

Healthcare is improving every day because of emerging technologies such as mobile applications, sensors, IoT, and wearable devices. Today, sensors can be embedded in almost everything in our daily life, such as clothes, watches, belts, or even glasses. These sensors can be used for motion tracking and vital signs measurement to send them to a smartphone for monitoring [6]. A survey classified the use of IoT in the medical sector into two categories: services (ambient assisted living, Internet of m-health, children health information and access to wearable devices, etc.) and applications (glucose level sensing, blood pressure and body temperature monitoring, etc.) [7]. The survey also covers the security requirements for an IoT-healthcare system, including:

- Confidentiality: It ensures the protection of medical data from being accessed by unauthorized parties. Only authorized users can access the data.
- Integrity: Its role is to ensure that the medical data received are strictly identical to those issued and that they have not been tampered with.
- Authentication: Is the process that enables a medical device to authenticate and identify the peer with which it is interacting.
- Availability: It ensures the availability of medical services and IoT devices anytime and anywhere for authorized parties.
- Non-repudiation: It ensures that a user or a medical device cannot deny being the transmitter of a message.
- Authorization: It ensures that only the authorized party can have access to the data or have the authorization to perform specific tasks.

In addition to these requirements, we should also add privacy concerns. Privacy is the right to protect personal data from others and malicious devices. As mentioned in [8], privacy should be protected at different levels: device level, during communication, at the storage and processing levels. The authors in [7] and in [9] also highlight some security challenges that IoT poses in healthcare, such as:

- Computational, memory, and energy limitations: The IoT devices have in general low speed processors, low on-device memory, and limited battery power.
- Mobility: By nature, IoT medical devices and wearables are mobile, thus they can be connected to the Internet through different networks, which makes them vulnerable to attacks and virus spreading.
- Access control and data leakage.

2.2. Blockchain

2.2.1. Blockchain overview

Blockchain technology is highlighted, for the first time, by Satoshi Nakamoto, who created the first cryptocurrency called Bitcoin [10]. Blockchain is a technology that allows both transactions storage and transmission. It stores data in a ledger made up of a set of blocks. Each block is linked to the previous one to construct a chain of blocks. Data transmission is ensured through a peer-to-peer network. Hence, Blockchain is a distributed ledger shared in a secure and decentralized manner. Over the last few years, Blockchain gained tremendous attention in the finance and banking sectors. Nowadays, it is making its way into other areas of application such as insurance, energy, industry and healthcare. Thus, Blockchain becomes so popular in almost all domains due to its features: it is decentralized, distributed, and secured. The network is decentralized; thus, there is no need for a central authority to govern the network. Data is archived using a consensus algorithm to reach agreement between nodes. The ledger is distributed, and it is maintained by all nodes involved in the network.

2.2.2. Blockchain security

The security in Blockchain is ensured by many elements:

- Cryptography: In general, Blockchains use public-key cryptography (known also as asymmetric cryptography). This kind of cryptosystem offers several security mechanisms namely; key establishment, encryption, decryption, digital signature, and identification [11]. Blockchain uses the asymmetric cryptography system to create a pair of keys (public and private). These keys are used to identify and authenticate users' accounts. The public key is derived from the private one, and it is used to create an address. The last one is used as a unique identifier of its owner in the network. The private key is used for a personal digital signature to prove the ownership of an account and to ensure that the message was signed by the owner of a corresponding private key (see Fig. 1).
- Consensus algorithm: is the mechanism used to reach agreement. Bitcoin, for example, uses the Proof of Work (PoW) consensus algorithm which is based on using computing power to solve a complex mathematical problem to find the block hash.
- Immutability: Blockchain is an append-only system. All blocks are linked by a cryptographic hash which prevents Blockchain's tampering.
- The replication of data across all nodes due to its distributed nature. Thus, in Blockchain, there is no single point of failure.
- Traceability: Blockchain allows the recording of a full and timestamped history of all transactions.

Blockchain is split into three types: public, private, and consortium. The difference is related to data visibility and the consensus process. In public Blockchain, everyone can participate in the consensus process, have access to the ledger and append data to it. In private Blockchain, both access to the ledger and participation to the consensus process need permission from the owner organization. The consortium Blockchain groups both types. The access to the ledger can be public or private. However, the consensus process can be controlled by several organizations instead of a single one.

2.2.3. Ethereum

Ethereum [12] is an open source Blockchain-based platform with decentralized and distributed computing. It was created by Vitalik Buterin in 2014, inspired by the cryptocurrency Bitcoin. Like in Bitcoin, Ethereum uses Elliptic Curve Digital Signature Algorithm (ECDSA) which was standardized in FIPS 186–4 [13]. Elliptic curve cryptography is based on the discrete logarithm problem to create a pair of keys. The elliptic curve used in Ethereum is called *secp256k1* and is determined by the following function [14]:

$$y^2 \bmod p = (x^3 + 7) \bmod p \tag{1}$$

where p [1] is a constant and the mod p is to indicate that this curve is over a finite field of prime order p. The variables x and y are the coordinates of a point on the *secp256k1* curve. The creation of a private key consists of picking a random number between 1 and 2^{256}. The public key is derived from the private one using the elliptic curve multiplication: $K = k * G$ where K is the public key generated from the private key k and G is the generator point, which is a constant point.

When a transaction is sent to the Ethereum network, It is signed by the user digital signature generated by his private key. On the other hand, anyone on the network can check if the transaction is valid. The validation process is done by verifying that the digital signature matches the Ethereum address, so the public key, of the sender.

The purpose of Ethereum is to provide Ethereum virtual machines (EVMs) that can run computer programs named "Smart contracts". The term "Smart contract" was used for the first time by Nick Szabo in 1997 [15]. Smart contract combines a set of protocols with user interfaces to formalize and secure relationships over computer networks. In Ethereum, smart contracts are computer programs that are deployed and stored in the Ethereum Blockchain network to run by the EVM. These programs constitute the translation of a contractual logic according to some rules and regulations whose output is agreed upon by the network nodes. They can be self-executed, without the intervention of a third party, when the predefined conditions are achieved. A smart contract can contain data, send transactions, exchange values, or assets, and communicate with other smart contracts. Another purpose of Ethereum is the creation and deployment of decentralized applications (Dapps). The Dapp front-end is similar to the front-end in any other traditional application. However, the difference between the two types is in the back-end side. In Dapp, the back-end is developed using Blockchain smart contracts. Thus, instead of being implemented in a central server, the back-end is running on the Blockchain network and is building using smart contracts.

2.3. IPFS

IPFS (InterPlanetary File System) [16]: Is a peer-to-peer distributed file system. The new with IPFS is that it has replaced the location-based addressing by content-based addressing. In other words, to search for some data, we need its hash to request for it instead of the address at which it is located. When a file is sent to the IPFS for storage, a unique hash is generated for it. Thus, to look up this file, you just need to look up its hash. Fig. 2 explains how to send/get data in IPFS.

3. Related work

As it handles critical information, healthcare is one of the most sensitive domains that requires a big interest in terms of privacy, security, access control and availability of medical records. The security in such a sector must be guaranteed all over different stages: transmission, storage and manipulation. In this section, we present some works that processed the security of medical records using Blockchain technology.

In [17], the authors proposed an approach to manage permissions and control access to medical records using smart contracts. In their prototype implementation, they used the Ethereum Blockchain network to deploy their smart contracts that allow patients to control and manage access to their data. Data storage is done in a Gatekeeper database. Another work proposed to integrate Blockchain with mobile applications to share data with healthcare providers and insurance companies [18]. They used Hyperledger Fabric as a permissioned Blockchain, where they stored policies for access control and for integrity reasons. Ancile is another proposed framework for access control management and increasing interoperability in the case of electronic health records. This framework is similar to the previous one, but is based on a permissioned Ethereum-based Blockchain (called Quorum) instead of Hyperledger Fabric [19]. All these solutions did not integrate in their systems the use of IoT devices. The use of this kind of devices becomes a necessity due to the benefits that are offered in terms of remote medical monitoring. Especially when we treat diseases that require regular check-ups and long time treatment. The authors in [20] proposed a novel Blockchain framework to preserve privacy in healthcare IoT devices. They used lightweight cryptographic techniques to encrypt data and validate transactions. Although, they did not implement the system yet. Another work also considers

[1] $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

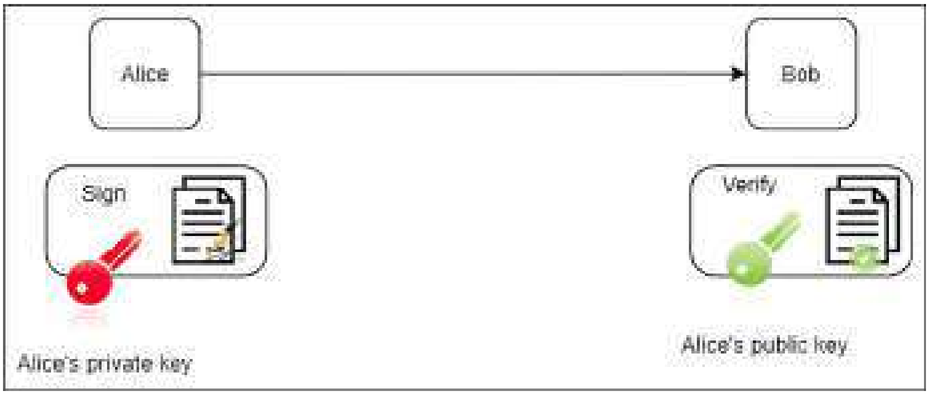


Fig. 1. Public key digital signature scheme: Alice uses her private key to sign her messages. Bob can verify that the message was signed by Alice using Alice's public key.

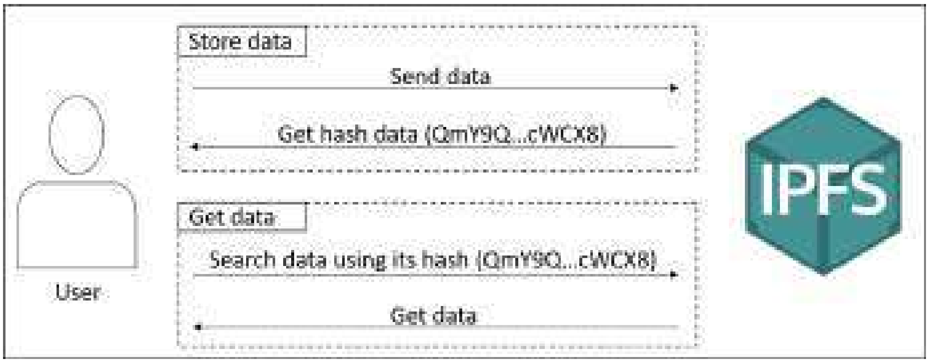

Store data
Send data
Get hash data (QmY9Q...cWCX8)
Get data
Search data using its hash (QmY9Q...cWCX8)
Get data
User
IPFS

Fig. 2. Store and get data from IPFS.

the use of IoT with Blockchain to create a medical platform to ensure EMR integrity management [21]. In this work, they used Hyperledger for testing. Mamta et al. [22] proposed a scheme for online storage and retrieval of personal health data in a typical cloud-based cyber-physical systems. Their solution is based on the use of blockchain and attribute-based searchable encryption.

All these solutions are using a centralized database to store data and not all of them are encrypting data before storage. Various other researches have proposed systems and approaches that combine IoT with Blockchain technology for security reasons. Lu et al. [23] designed a group signature scheme based on the SPS-EQ signature and PoK suitable for resource-constrained sensors, then proposed a novel blockchain-based cloud storage protocol for sensors in industrial IoT. In [24], the authors propose to use both private and public Blockchain to secure IoT devices. However, they eliminated the PoW in their architecture. This speeds up the blocks appending to the Blockchain because there is no need to solve the PoW. Another work proposed to use a LoRa gateway for routing data from IoT devices to an Ethereum Blockchain [25]. However, this work is still in progress, and there is no system implementation yet. The authors in [26] proposed a healthcare system design framework based on Blockchain and IoT. In [27], the authors proposed using Ethereum to manage the access to medical data using smart contracts. A gateway is used to handle data generated by medical sensors and an off-chain database based on IPFS is used to store data. However, also these two last works did not present the system implementation details. In a recent work, the authors proposed a system based on Blockchain and IPFS combined with attribute-based encryption to store and secure EMR [28]. However, in their work, they did not integrate IoT devices and the implementation of the system has not yet been completed. Another work proposed a secure intrusion, detection using Blockchain with a classification model for cyber-physical systems in healthcare [29].

4. A secure healthcare system based on IoT, Blockchain and IPFS

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

4.1. System architecture

As shown in Fig. 3, the proposed system architecture has three sides: patient side, medical team side and IPFS side. The patient and medical team sides are linked together by a Blockchain network and both of them can communicate with the IPFS side. Health data is encrypted and stored in the IPFS, while the Blockchain stores a hash link to this data.

4.1.1. Patient side

It has two categories of devices:

- IoT medical devices: each patient has a set of IoT medical devices and electronic wearables with embedded sensors. These sensors are able to capture different vital signs measurements and track physical activity and sleep patterns, etc. Thus, the role of these devices is the collection of health data to share them with the medical team side through the patient's smartphone. Each of these devices is registered on the Blockchain by its owner (patient) and is identified in the network by a pair of unique values: its MAC address and the identifier of its owner.

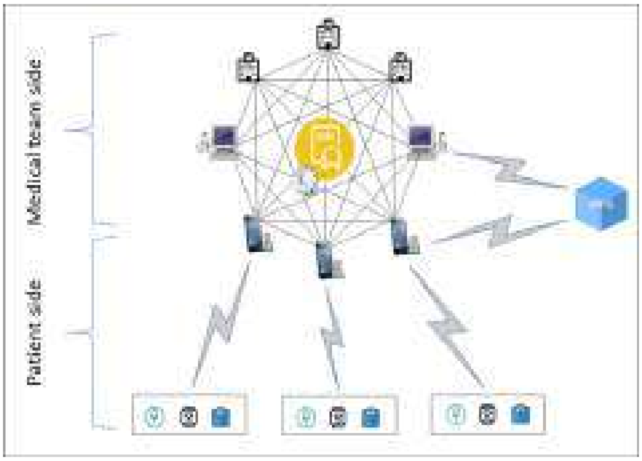


Fig. 3. Architecture design of our system.

- Smartphone: it is the intermediate device between IoT medical devices and the medical team. Thus, it is the node that allows other IoT devices to access to the Blockchain. Due to its limited capabilities, the smartphone will not store the whole Blockchain but will just have access to through a Dapp. When the patient creates his profile, the Dapp creates a Blockchain account for him, and a pair of keys is generated to be the unique identifier for him.

4.1.2. Medical team side

It includes physicians, hospitals, pharmaceutical laboratories and public health organizations. These entities are connected with patients through a Blockchain network to have access to their health data. Physicians are connected to the Blockchain, as a light node, through their computers or smartphones. Hospitals play the role of full nodes that can store a copy of the Blockchain and participate in the consensus process. The other entities can have a copy of the Blockchain, but they cannot participate in the consensus process. The patient data will be used for remote monitoring, analysis, and research purposes.

4.1.3. IPFS side

On this side, we use the IPFS, which is a peer-to-peer distributed file system, as an off-chain database to store encrypted health data. We used IPFS for data storage instead of Blockchain for two reasons, in one hand, because the system handles a large amount of data which are generated by a large number of devices. Storing such an amount of data in the Blockchain can affect its size and will need special full nodes to store data. Thus, the Blockchain is used only for access control and to protect data integrity. On the other hand, since it is not practical to store sensitive data in the Blockchain even if it is encrypted. As we know, data are stored in the Blockchain eternally, so if the encrypted system is broken in the future, all nodes can then decrypt and have access to the data of all patients in the system.

4.2. Proof of authority

In this work, we used the Clique PoA [30] as a consensus algorithm for the benefits it provides. Clique is an Ethereum implementation of the PoA. It is implemented by the Geth Ethereum client [31] and is based on the Go language. PoA is one of the most popular consensus algorithms used in permissioned Blockchain to reach agreement and secure the network. The use of such an algorithm provides many advantages. On the one hand, it can speed up the consensus process, so it can accelerate data storage. This aspect is essential in healthcare applications that require real time processing. On the other hand, it doesn't require a huge computing resource which makes it more energy efficient. Contrary to the PoW algorithm, PoA allows only some predefined authorities, known as validators, to participate in the consensus process by validating transactions, creating and appending blocks to the Blockchain. In our case, these validators are hospitals. The PoA is specially 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 such as the ministry of public health.

4.3. System security

The security in the proposed solution is maintained by two aspects: the first one by benefiting from the Blockchain security features, and the second one by exploiting the proxy re-encryption mechanism.

4.3.1. Blockchain security features

As we already motioned in Section 2.2, the security of Blockchain is generally ensured by several elements, namely: Cryptography, Immutability, Replication, Traceability and Consensus algorithm. In addition to these features, our system exploits also smart contracts for access control and proxy re-encryption for data encryption. In this work, we integrated smart contracts to register or delete devices, grant or revoke access to data and verify the authentication.

4.3.2. Proxy re-encryption

The proxy re-encryption is a cryptosystem scheme proposed for the first time in 1998 [32]. It refers to a mechanism that allows a tired party (a proxy) to transform Alice's cipher text by encrypting it to be decrypted by Bob. The most important thing in this mechanism is the fact that the proxy has access to the cipher text only but not to the plain text. It then encrypts the cipher text with Bob's public key, and it does a re-encryption without knowing the plain text. The re-encrypted text can then be decrypted by Bob using his private key. In our approach, Alice is represented by the patient, and Bob is represented by the physician. A proxy re-encryption scheme is defined by the following five algorithms [33]:

- KeyGen: This algorithm outputs a pair of keys (public key pk and private key sk) for a given security parameter $k \in K$on input.
- ReKey: This algorithm is used by user i to generate a re-encryption key $rk_{i \to j}$ for another user j. To do so, the algorithm takes on the input the key pair for user i (pk_i, sk_i) and the key pair for user j (pk_j, sk_j) (but sk_j is optional). User i is the owner of the message and user j is the user for whom the message will be sent.
- Encrypt: For a given plaintext message $m \in M$, user i uses his public key pk_i to encrypt it. Encrypt outputs an original ciphertext $c_i \in C_1$. Where C_1 is the set of original ciphertext.
- ReEncrypt: This algorithm takes as input the ciphertext $c_i \in C_1$ for user i and the re-encryption key $rk_{i \to j}$. This algorithm is used by the proxy to create a transformed ciphertext $c_j \in C_2$ for the user j, where C_2 is the set of transformed ciphertext.
- Decrypt: This algorithm is used to decrypt a ciphertext and it gives as output its corresponding plaintext message. For user i: It takes in input the private key sk_i and the ciphertext c_i. For user j: It takes on input the private key sk_j and the re-encrypted message c_j.

5. Use cases and practical scenarios

Before presenting the implementation of our architecture, in this section, we present a principal use case and some practical scenarios applied to a diabetes management case.

5.1. System functionalities

Once a patient installs our Dapp, he will not be able to create his identity until he receives a confirmation from his physician. Thus, the physician is the only person who has the right to register his patients by generating a QR code for each of them. In this QR code, we integrate the physician address and some other information about the Blockchain network such as its chainID. After scanning the QR code, the patient can create his account by generating an Ethereum address (a unique identification), therefore implicitly his private and public keys. Once the Blockchain network is reached, the patient will be able to perform several functions by using different Dapp interfaces. In addition, to verify the authentication, the smart contract is also the engine which controls our Dapp back-end providing the different functionalities presented in the use case diagram in Fig. 4.

These functionalities enable:

Patient to:

- Handle his devices (add or delete devices): To be able to send his first data, each patient has to register first all his medical devices that are used to collect different data. Each device will be identified on the Blockchain by a unique set of values that include its mac address and the address of its owner (patient Ethereum address) (Fig. 5) [34]. In this way, we can ensure that no one can add a device except its owner and that only data from registered devices are accepted. Patients can then be protected from malicious devices that could send wrong data in behalf of them, for example. After registering his devices, a pair of keys ($sk_{patient}, pk_{patient}$) is generated. The patient public key is then used to encrypt the data that is collected from the devices before being forwarded it to the IPFS through the patient's smartphone. The Dapp will also generate the re-encryption key and send it to the hospital. This last will be in charge to re-encrypt data for the eligible entities (physicians, researchers ...).
- Handle access to his data: Each patient can control access to his own data by granting access to eligible healthcare institutions and revoking it from others.

Physician to:

- Give his patients access to the network
- Have access to his patient data to follow his health status.
- Request for access when he treats a patient which not belonging to him in case of emergency. The system will give them temporary access to the necessary data.

Researchers and public health organizations to:

- Extract information to use it for research purposes, such as data analysis and statistics. These entities must have in advance the patient's consent to have the access right.

All data manipulations are stored in the Blockchain for traceability reasons.

5.2. Case study: diabetes management

Diabetes is a major chronic disease that has reached alarming levels. According to the International Diabetes Federation [35], in 2019, the number of people living with diabetes in the world is 463 million. This number is predicted to rise to 578 million in 2030 and 700 million in 2045. In 2019, the number of children and adolescents (up to 19 years old) living with type 1 diabetes is over one million. 136 million people with diabetes are over 65 years old. Almost 30% of diabetic patients are children and elderly age groups. In what follows, we give three possible diabetes assisting scenarios and how our solution could react in each case.

5.2.1. Hypoglycemic emergency in the case of an elderly person

Hypoglycemia is a state caused by low blood glucose levels. It is often known as a side effect of diabetes therapy. Many symptoms could accrue in case of hypoglycemia such as irregular heartbeat rhythm, weakness, nervousness, sweating, visual disturbances, loss of consciousness, or even coma in case of severe hypoglycemia. Elderly patients with diabetes are the most affected by severe hypoglycemia due to inappropriate intensification of therapy [36]. Let us consider Bob who is an elderly diabetic patient. Bob stays home most of the time alone. By using the proposed solution in Section 4, the medical team could easily track his medical situation in a secure manner (Section 4.3). They could also detect and prevent possible hypoglycemic events by early recognizing its symptoms. The collected data are shared with eligible entities while preserving confidentiality and data integrity. Based on this data, the system could send notifications to the medical team to prevent a possible hypoglycemia event. The system would also send alerts to the emergency service in case of emergencies such as a hypoglycemia event, falls, loss of consciousness or coma.

5.2.2. Person having an emergency while traveling

Alex is a diabetic man who travels all the time due to his work. By using our system, he could manage his diabetes even when he travels by caring his medical and wearable devices. These devices could collect data and send it to his smartphone which will be in charge of sharing it with his medical team for remote monitoring. In case of an emergency, the system would be able to send an alert to the nearest emergency service to instruct an ambulance to assist the patient. The system will check if the emergency physician is enrolled in a known medical entity. If so, the data will be re-encrypted for them, and the system will give the physician a temporary access (Section 5.1). The physician will be able to get the patient data just during the treatment, but he will not be able to access to it in the future.

5.2.3. Assisting diabetic children

Alice is a 6 years old girl with type 1 diabetes. Her tutor could use our solution to track her health data and monitor her diabetes by using his smartphone. Alice should use her medical wearables for data collection. Her tutor's smartphone could then get data from these wearables once Alice is back home.

6. Implementation

In this section, we present the proposed solution implementation. The system is implemented using a private Ethereum Blockchain network based on Clique PoA, IPFS for data storage and proxy re-encryption for data encryption.

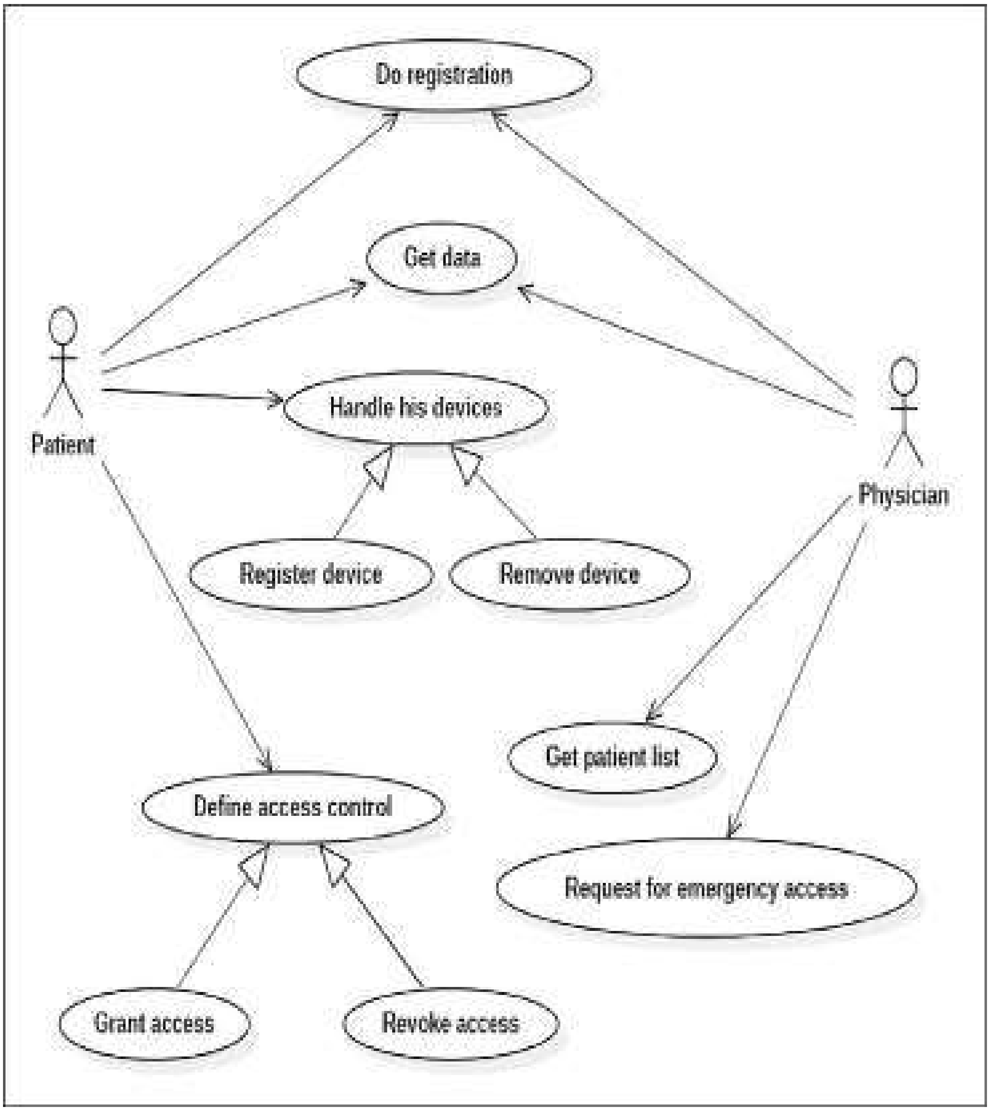


Fig. 4. Use case diagram.

6.1. Node types

As already mentioned, we used the Geth client implementation created by Ethereum. In the proposed solution, we have three types of nodes:

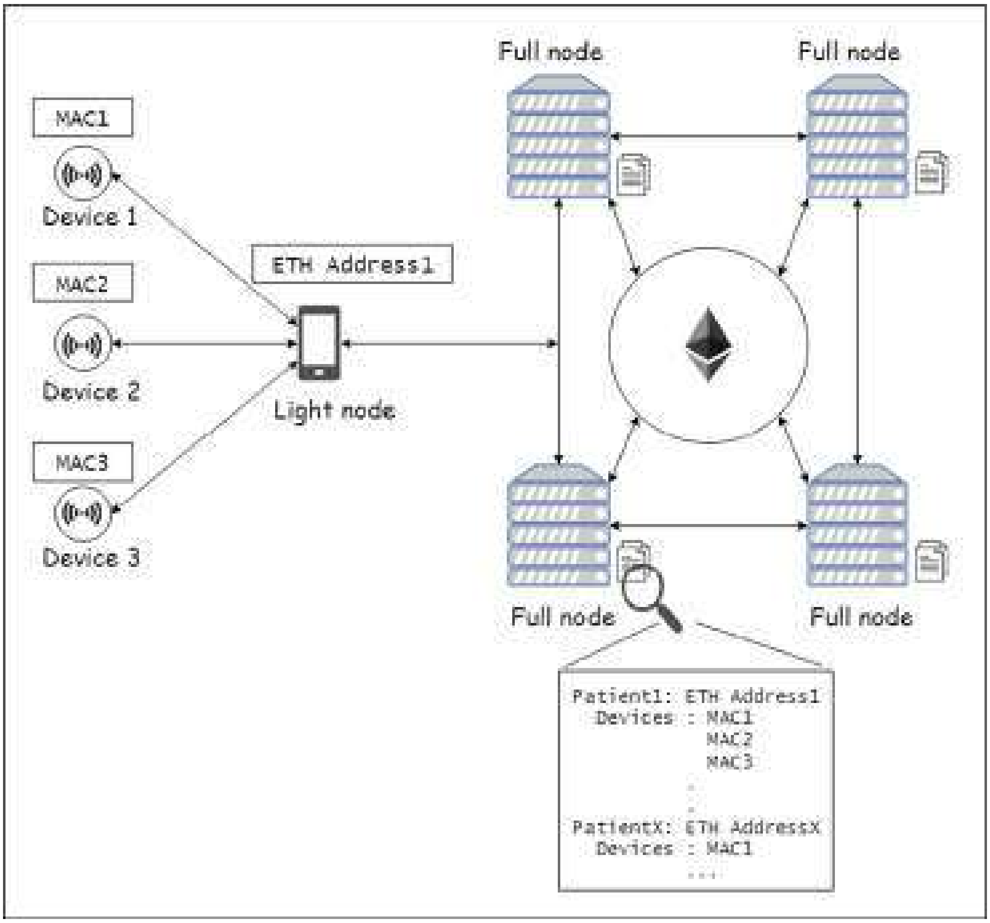
- Patient nodes: are patients' smartphones. These nodes have access to the Blockchain through the patient Dapp which is using the Web3j API [37] to interact with our Ethereum Blockchain network. The connection with the network is established using JSON-RPC protocol.
- Physician nodes: can be either physicians' computers or smartphones. The second type is similar to the patient nodes. The difference is that the connection with the Blockchain network is ensured by the physician Dapp instead of the patient Dapp. Like the first, it is hosting light Ethereum clients [38].
- Hospital nodes: are nodes belonging to hospitals and public health organizations. They are full Ethereum nodes that store the whole Blockchain and participate in the consensus process to validate transactions and create new blocks. These nodes are also participating in a private IPFS network in which we store the encrypted data.

6.2. Data collection

In the case of a diabetes patient, there are many healthcare parameters that are necessary for diabetes management. Among these, we have to monitor glucose and A1C levels; cholesterol and triglyceride levels; blood pressure, body weight, and activities. For the test, we will treat some of these parameters, as shown in Table 1. A Raspberry Pi will be used to simulate a glucometer. The patient could insert his weight manually and upload his health assessment using the Smartphone. Activities will be tracked using the Smartphone sensors.

6.3. Data encryption/ decryption

Data encryption/decryption is ensured by the Umbral threshold proxy re-encryption scheme created by NuCypher [39]. In our work, we used pyUmbral, which is the python reference implementation of Umbral. In this work, we develop mobile Dapps using Android and Ethereum. The integration of pyUmbral into our Android code is done using a python SDK for Android named Chaquopy [40]. This SDK enables the use of Python code on Android applications. Fig. 6 explains how the Umbral threshold proxy re-



```
Patient1: ETH Address1
  Devices : MAC1
            MAC2
            MAC3
            .
            .
PatientX: ETH AddressX
  Devices : MAC1
            ...
```

Fig. 5. Devices identifier.

Table 1
Health parameters used in our work.

Devices	Parameters
Raspberry pi Smartphone	Blood glucose Weight Activities Health assessment

encryption is implemented in our approach. First, we generate a pair of keys (public and secret keys) for both the patient and the physician. The patient encrypts the data using his public key ($pk_{patient}$). Then, generates the re-encryption key (p_{rek}) by using his secret key and the physician public key. Once the re-encryption key is generated, it is sent to the hospital (the full node), which will play the role of a proxy that will re-encrypt the data on behalf of the physician. The hospital checks if the physician has permission to access data. If the physician has access rights, the hospital re-encrypts data for him by using the re-encryption key

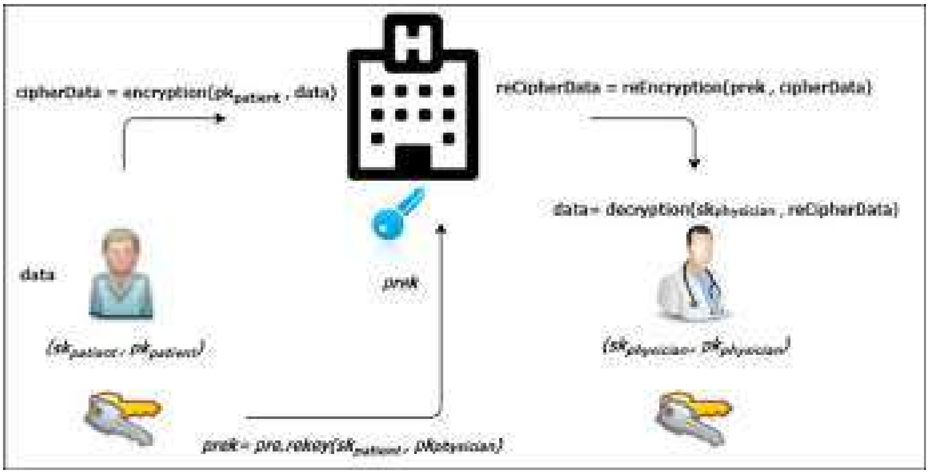

cipherData = encryption(pk_patient , data)
reCipherData = reEncryption(prek , cipherData)
data= decryption(sk_physician , reCipherData)
data
prek
(sk_patient , pk_patient)
(sk_physician, pk_physician)
prek= pre.rekey(sk_patient , pk_physician)

Fig. 6. Implementation of proxy re-encryption in our architecture.

without having access to the plaintext. The physician will then be able to decrypt the re-encrypted data by using his secret key ($sk_{physician}$).

6.4. Data storage

Once the data is encrypted using the patient public key, our Dapp sends it to the IPFS for storage. IPFS then generates a unique hash for each data. When the hash is received, the Dapp stores it in the Blockchain. To interact with the IPFS nodes, we used the java-ipfs-http-client [41] which is a Java client for the IPFS HTTP API. The IPFS network is deployed on Ubuntu 18.04 nodes. IoT devices are simulated by a Raspberry Pi Model 3B. Fig. 7 represents the steps of the data storage process in our approach.

6.5. Smart contract functions

In this work, we developed a smart contract with different functions to ensure different system functionalities using Solidity language. Fig. 8 illustrates the fundamental interactions with the smart contract in the ordinary scenario. The described process consists of 10 steps:

- Step 1: Each physician adds and registers his patient in the Blockchain.
- Step 2: Once the patient is registered, he will be able to register his devices in the Blockchain using his smartphone and the Dapp interface.
- Step 3: The patient grants access to his physician so that he can access the data.
- Step 4: When the patient wants to send his health data, the Dapp encrypts them then sends them to the IPFS.
- Step 5: The IPFS sends the data hash back to the Dapp.
- Step 6: The Dapp stores data hash in the Blockchain.
- Step 7: When the physician wants to get data, he sends a request through the Blockchain.
- Step 8: After verifying the physician's access permission, the smart contract sends confirmation to the hospital.
- Step 9: The hospital gets the encrypted data from the IPFS.
- Step 10: The hospital re-encrypts data and sends them to the physician.

Each request sent to the Blockchain is processed by the smart contract, which authenticates and verifies the eligibility of the requester.

Next, we give the pseudocode of some of these functions.

Algorithm 1 is used to register a patient in the network. It takes as parameters: the patient name, his phone number, and the address of his associated physician.

Algorithm 1: Register a new patient

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

Algorithm 2 is the function to invoke whenever a patient wants to register a new device in the Blockchain. This function takes in parameters the device MAC address. It adds a device to the list of the patient who sent the transaction. It links the device MAC address with the patient address (which is represented in the algorithm by the sender). Before adding the device, it first checks its existence in the patient list. Thus, we can prevent the registration of the same device many times, especially in the case of malware infection.

Algorithm 2: Register a new device

```
procedure adddevice (sender, macAddress)
  // Append the Mac address to the sender's list of devices
  $require(!deviceExists[macAddress])$
  $devicesList[sender].push(macAddress)$
  $deviceExists[macAddress] = true$
end procedure
```

Algorithm 3 represents the function responsible for adding a permission role to a physician. The function takes in parameters, the physician address for whom the permission will be accorded, the read and write permissions, and the hash information. The last one is used in the re-encryption process. For each patient, we create a list of entities for which he accorded the access and permission given for each of these entities.

Algorithm 3: Add Permission

```
procedure addpermission (sender, physicianAddress, read,
  write, hashInfo)
  $P \Leftarrow Permission(physicianAddress, read, write, hashInfo)$
  $permissionList[sender].push(P)$
end procedure
```

Algorithm 4 is describing the function which is responsible for storing the hash data in the Blockchain. It takes in parameters, the hash data, the identifier of the device producing this data and a capsule. This last parameter is the information that will be used later for decryption.

Algorithm 4: Add data hash

```
procedure adddata (sender, device, dataHash, capsule)
  for $(i = 0;\ i < devicesList[sender].length;\ i++)$
    if $(devicesList[sender][i] = device)$ then
      $record \Leftarrow EMR(device, dataHash, now, capsule)$
      $EMRs[sender].push(record)$
    end if
  end for
end procedure
```

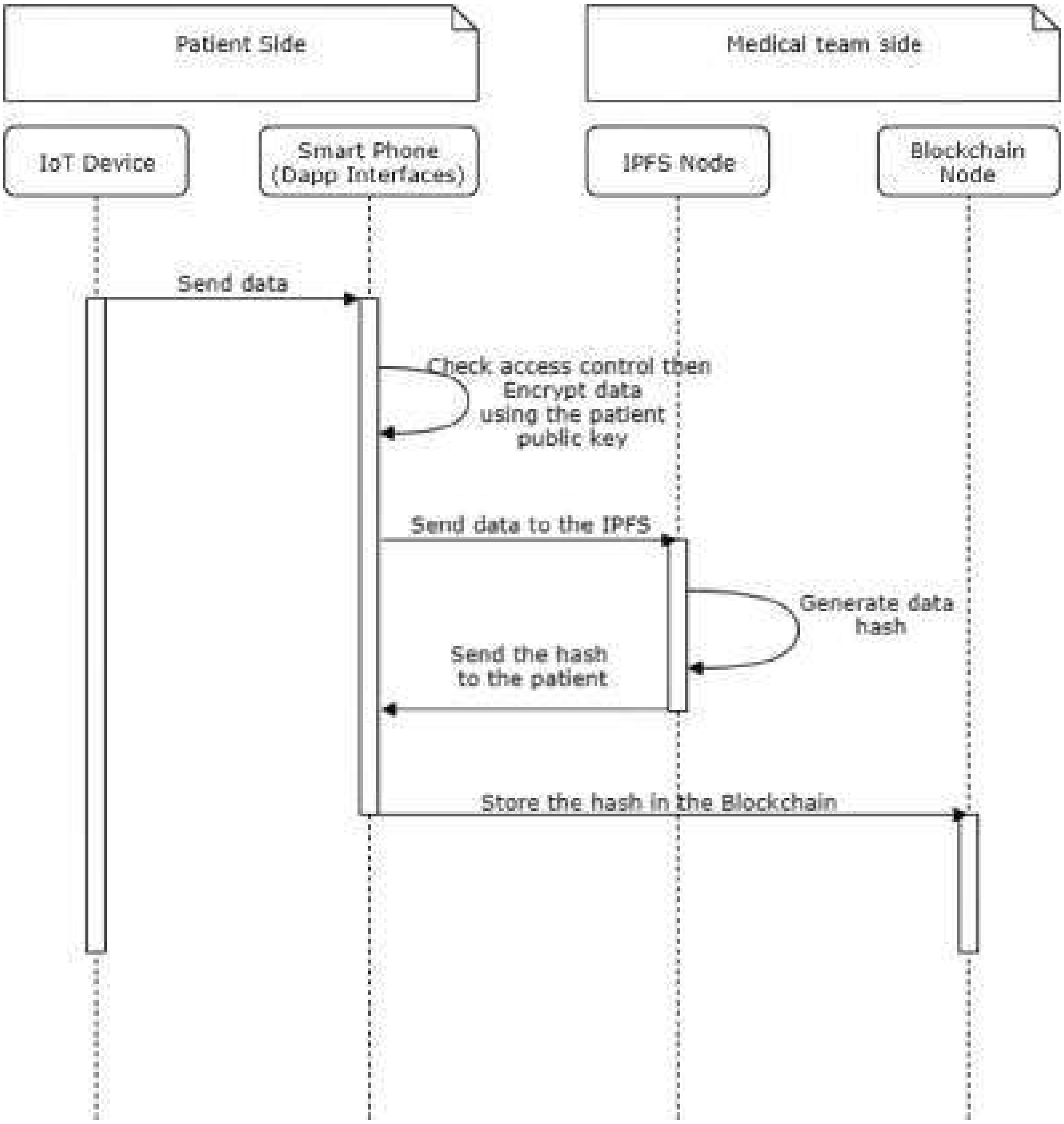

Patient Side
Medical team side
IoT Device
Smart Phone (Dapp Interfaces)
IPFS Node
Blockchain Node
Send data
Check access control then Encrypt data using the patient public key
Send data to the IPFS
Generate data hash
Send the hash to the patient
Store the hash in the Blockchain

Fig. 7. Sequence Diagram for data storage.

6.6. Execution results

In this section, we present screenshots of patients and devices registration forms from the Android Dapp and the result of invoking some functions using Remix IDE [42]. Remix is an IDE that allows the creation, compilation, and deployment of smart contracts. It also provides an environment to execute transactions and interact with the smart contract. First, we create a private Ethereum Blockchain using five nodes. We create and deploy our smart contract using Remix. To interact with this smart contract, in addition to the android Dapp, we also use the Web3 Provider to connect the Remix with one of our Blockchain nodes (see Fig. 9).

In Fig. 10, we show a registered patient information using Remix.

In Fig. 11 we show the different functionalities of our application. The first item is to scan for new devices and show the list of paired ones. The Encryption item is to show the patient address and his QR code. The profile item is used to manage patient information. Send Data item is used to share data (weight, activities, health assessment) with the physician, and Get Data is used to get patient data.

Fig. 12 illustrates devices handling, Fig. 12a represents the list of paired devices with Bluetooth. Once the patient registers his device, the *addDevice* function adds this device to the list of devices of this patient. Thus, in this way, no one can add a device to the list of a patient except himself.

In Fig. 12b, we show the device located at index zero of patient1. Thus, each device is linked to its patient address and its MAC address.

7. Discussion

In this section, we explain how the proposed solution ensures scalability and low processing time. We discuss also the principal security goals ensured by our approach and compare it with the existent solutions.

7.1. Processing time and scalability

Processing time and scalability are affected by the consensus algorithm and the type of data storage used. The scalability of a consensus algorithm refers to the transaction throughput that an

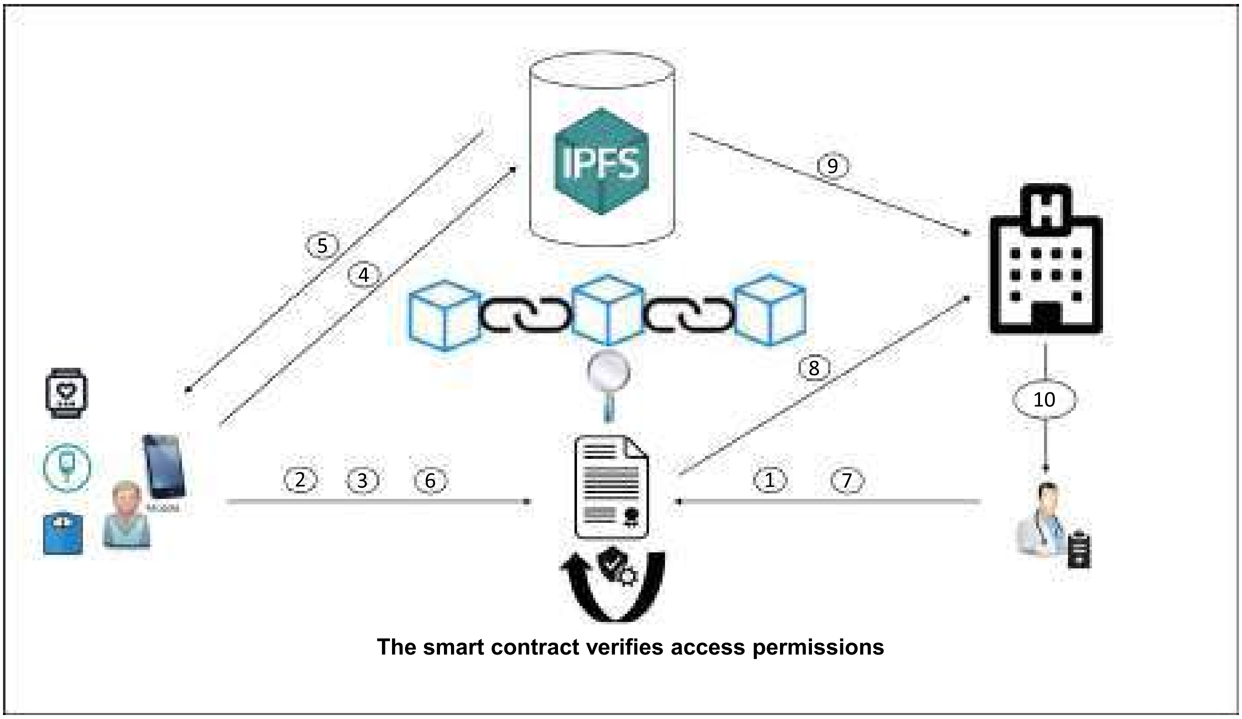


Fig. 8. Patient-Physician interactions with the smart contract.

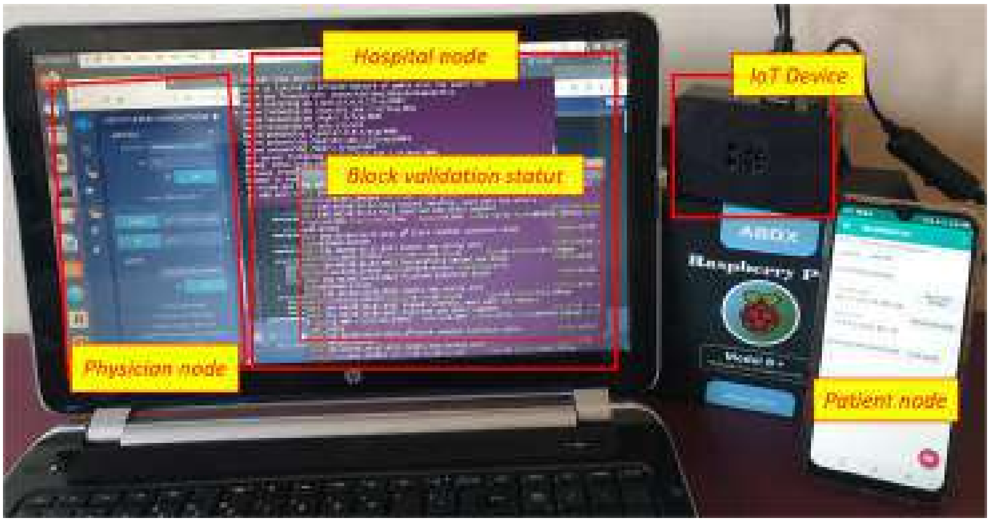


Fig. 9. The system implementation on PC, Raspberry and Smartphone.

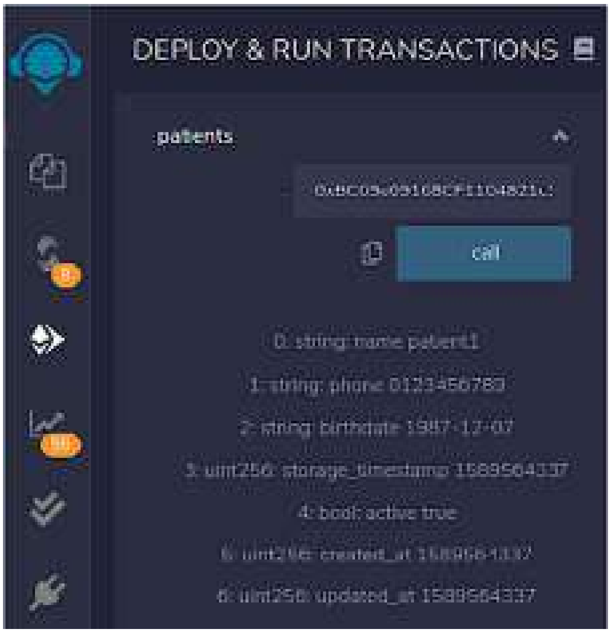
algorithm offers. Since the PoA is based on a limited number of validators, by using it we can achieve high throughput. Data scalability is also ensured by storing data hash in the Blockchain instead of the data itself. A low processing time is resolved by the use of PoA. The transaction time in PoA is faster than the transaction time in a network based on the PoW consensus. In our case, the block creation period is set to 5 second and the gas limit to 4700000. With this configuration, we can process 45 transactions per second (TPS). The processing times for some operations are illustrated in Table 2. PoA is more suitable for permissioned Blockchains and it increases the performance in terms of the validation processes time and the number of validated transactions per second.

7.2. Security analysis

7.2.1. Security model insurance

.

- Security and availability: Thanks to its security features, Blockchain is combined with IoT devices and data encryption to strengthen the security in our system. One of the most important features of Blockchain is the decentralization and replication of data. That prevents the single point of failure and ensures data availability. The availability in our system also benefits from the use of different medical devices, smartphones, and the Dapps that guarantee the interaction with the system.



```
0xBC09e09168CF1104821c
0: string: name patient1
1: string: phone 0123456789
2: string: birthdate 1987-12-07
3: uint256: storage_timestamp 1589564337
4: bool: active true
6: uint256: created_at 1589564337
6: uint256: updated_at 1589564337
```

Fig. 10. Read patient information using Remix.

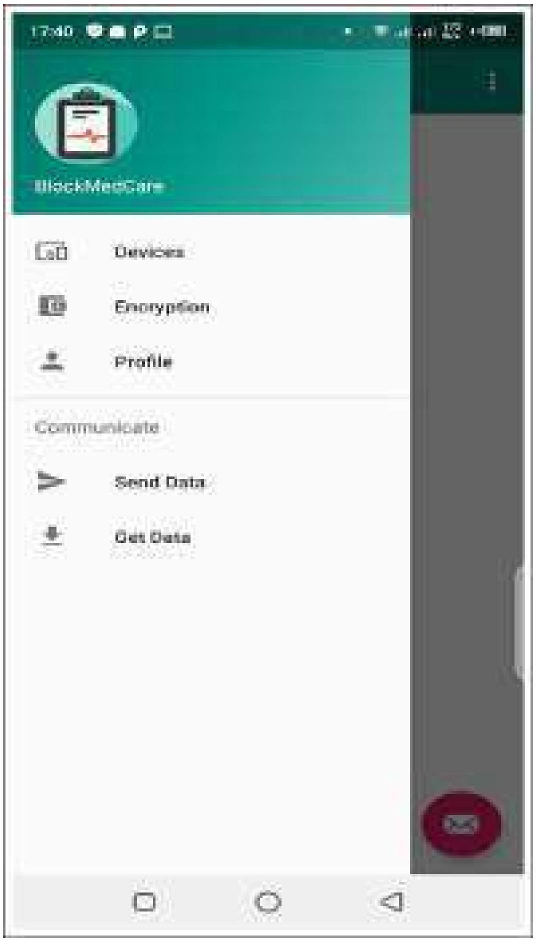


Fig. 11. BlockMedCare functionalities.

- Privacy: The preservation of a patient's privacy is guaranteed by using different concepts: The first is exploitation of the anonymity feature of the Blockchain. Thus, each patient is identified just by a unique identifier which makes his identification impossible by other nodes and restricted for his physician. The second one is data encryption, data is encrypted and stored in the IPFS and just his hash is stored in the Blockchain. The third one is controlling access to this data.
- Confidentiality: The confidentiality of patient data is ensured in our system by using smart contracts that are responsible for controlling access to this data. Thus, medical data are protected and could not be accessed by unauthorized parties.
- Integrity: Data integrity is enhanced in our system by using Blockchain technology. The Blockchain is used, on the one hand, to store hash data. This hash could not be changed once being registered due to the immutability feature which protects the data from being tempered. On the other hand, smart contract deployed in the Blockchain control access to the hash data. Thus, only authorized parties are able to access and append data. Blockchain also allows traceability by registering all manipulations and actions made in the data.

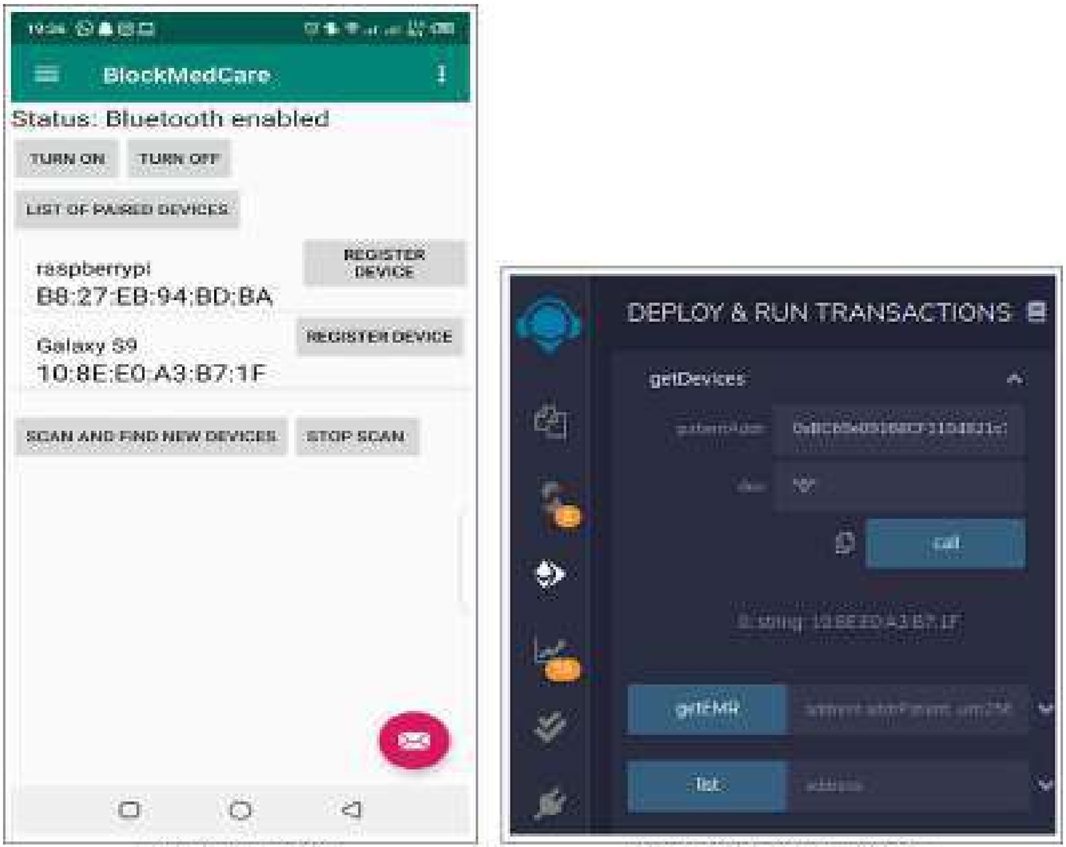
7.2.2. Resistance to attacks

.

- DDoS attacks: The distributed denial of service attack consists of disrupting the normal traffic of a service by flooding it with requests until not being able to receive further requests. The proposed solution is totally based on a decentralized and distributed system by using Blockchain and IPFS. This can protect against DDoS attacks.
- Impersonation attacks: In this kind of attack, an adversary impersonates the identity of a trusted entity. The difficulty of solving the elliptic curve discrete logarithm problem is due to the difficulty of solving the ECDSA.
- Message forgery attack: As its name indicates, it is used to forge or alter a message. This attack is also not possible because each transaction in the network is signed and verified before being stored in the Blockchain.
- Man-in-the-middle attacks: This attack consists of intercepting the communication between two entities without having a doubt that the communication between them has been compromised. In our approach, the transaction is encrypted and signed by a private key which is known only by its owner. To be validated, the transaction should be signed by a valid signature belonging to the emitting address. Thus, it is too hard for an attacker to forge a signature without knowing its associated private key.

7.3. Comparison with the existing works

In this part, we give a comparison between the proposed work and the most notable researches that consider the use of Blockchain technology to share and secure medical data. This comparison is based on the Blockchain type, the consensus mechanism, and the used network type (see Table 3). It also gives information about the data storage used in each work and whether it is taking into account the integration of IoT devices into its system or not. It can also gives information about the security level that each architecture provides and if it encrypts data before storing it or not. There are only two works ([17,43]) that are using the PoW consensus algorithm. The main problem with this algorithm is that it requires large amounts of energy to validate a block. It also can be considered as slow when it comes to the transactions speed. Furthermore, it did not take into account data encryption nor the use of IoT medical devices. In terms of data storage, all the cited



(a) Register a new device

(b) Read device information using Remix

Fig. 12. Devices Handling.

Table 2
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

works use a centralized database for data storage. This kind of storage is vulnerable to DDoS attacks and to be tampered. In contrast, in the proposed approach, we used IPFS for data storage to benefit from its decentralized storage nature. Moreover, our solution takes into consideration the essential requirements of security namely confidentiality, integrity, privacy and access control while other works are concentrating just on some of these requirements. In addition, most of the cited works which are based on IPFS for storage are not yet implemented.

Table 3
Comparison of our system with the existing works.

Name	Blockchain	Consensus	Network Type	IoT	Data storage	Data Encryption	Security Considerations	Implemented
MedRec [17]	Ethereum	PoW	Permissionless	No	centralized DB	No	Authentication, confidentiality	Yes
[18]	Hyperledger	PBFT	Permissioned	Yes	centralized DB	No	Integrity, privacy	Yes
Ancile [19]	Quorum	QuorumChain	Permissioned	No	centralized DB	Yes	Privacy, access control	Yes
[20]	Ethereum	PoA	Permissioned	Yes	centralized DB	Yes	Confidentiality, integrity	No
[21]	Hyperledger	PBFT	Permissioned	Yes	centralized DB	Yes	Confidentiality, integrity, privacy	Yes
							integrity, privacy	Yes
[43]	Not specified	PoW	Permissioned	No	IPFS	No	privacy, integrity	Yes
[44]	Not specified	Not specified	Permissioned	Yes	IPFS	Yes	privacy, access control	No
[45]	Ethereum	Not specified	Permissioned	Yes	IPFS	Yes	privacy	No
Our system	Ethereum	PoA	Permissioned	Yes	IPFS	Yes	Confidentiality, integrity, privacy, access control	Yes

8. Conclusion and future work

In this paper, we presented a healthcare system based on IoT, Blockchain, and IPFS technologies for managing chronic diseases. This system is offering a set of benefits for remote patient monitoring. It provides daily data collection, data sharing and security. The system is divided into three sides. The first side is responsible for data collection, this side is using IoT healthcare devices to ensure the collection. The second side is responsible for sharing data securely, the technology which is ensuring that is Blockchain. The last side is for data storage and it is using IPFS for that. Our system can be applied to any healthcare system. However, in our case, we choose to apply it especially to chronic diseases system management, because this kind of diseases requires daily follow-up and regular check-ups to be managed. The proposed system is fully decentralized, and it offers a high security level by using Blockchain, smart contracts, proxy re-encryption and IPFS to control access to patient data, protect privacy and ensure data integrity. As an example of application, we applied the system to a diabetes case and showed the execution results based on system interfaces and interactions with the smart contract through the Remix IDE. The comparison with the state of the art methods shows a good improvement of healthcare systems in terms of security.

In a future work, we aim to extend this work by implementing our solution using the Hyperledger Blockchain and compare it with the current solution based on Ethereum. We plan also to integrate artificial intelligence into our system to make it smarter and add more features. With artificial intelligence, we could provide data analysis, predictions, and prevention. The system would help physicians to make better clinical decisions and to provide an effective treatment.

Declaration of Competing Interest

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

Acknowledgment

This work is supported by the National Center for Scientific and Technological Research (CNRST).

References

[1] WHO. WHO — Integrated chronic disease prevention and control, online; accessed 05 July 2020; 2002. URL: https://www.who.int/chp/about/integrated_cd/en/..

[2] WHO. Technical package for cardiovascular disease management in primary health care. URL: https://apps.who.int/iris/handle/10665/252661. Online; accessed 19 December 2020; 2016. URL: https://apps.who.int/iris/handle/10665/252661..

[3] WHO. Global report on diabetes. WHO Press, World Health Organization: Geneva; 2016. oCLC: 948336981..

[4] Mobasheri MH, King D, Johnston M, Gautama S, Purkayastha S, Darzi A. The ownership and clinical use of smartphones by doctors and nurses in the UK: a multicentre survey study. BMJ Innov. 2015;1(4):174–81. doi: https://doi.org/10.1136/bmjinnov-2015-000062.

[5] Nguyen HH, Mirza F, Naeem MA, Nguyen M. A review on IoT healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. In: 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD). Wellington, New Zealand: IEEE; 2017. p. 257–62. doi: https://doi.org/10.1109/CSCWD.2017.8066704.

[6] Haghi M, Thurow K, Stoll R. Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices. Healthcare Inf. Res. 2017;23(1):4. doi: https://doi.org/10.4258/hir.2017.23.1.4.

[7] Riazul Islam SM, Kwak Daehan, Humaun Kabir M, Hossain M, Kwak Kyung-Sup. The Internet of Things for Health Care: A Comprehensive Survey. IEEE Access 2015;3:678–708. doi: https://doi.org/10.1109/ACCESS.2015.2437951. URL: http://ieeexplore.ieee.org/document/7113786/.

[8] SathishKumar J, Patel DR. A Survey on Internet of Things: Security and Privacy Issues. Int J Comput Appl 2014;90(11):20–6. doi: https://doi.org/10.5120/15764-4454. URL: http://research.ijcaonline.org/volume90/number11/pxc3894454.pdf.

[9] Adat V, Gupta BB. Security in Internet of Things: issues, challenges, taxonomy, and architecture. Telecommun Syst 2018;67(3):423–41. doi: https://doi.org/10.1007/s11235-017-0345-9. http://link.springer.com/10.1007/s11235-017-0345-9.

[10] Nakamoto S, Bitcoin. A Peer-to-Peer Electronic Cash System. Bitcoin 2008:9.

[11] Bashir I, Safari aOMC. Mastering Blockchain – Master the theoretical and technical foundations of Blockchain technology and explore future of Blockchain technology; 2017. oCLC: 1105777084..

[12] Ethereum. ethereum/wiki, online; accessed 10 December 2019; 2019. URL: https://github.com/ethereum/wiki..

[13] Information Technology Laboratory, Digital Signature Standard (DSS). Tech. Rep. NIST FIPS 186–4, National Institute of Standards and Technology (Jul. 2013). doi:10.6028/NIST.FIPS.186-4. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf..

[14] Antonopoulos A, Wood G. Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly Media, Incorporated; 2018. URL: https://books.google.co.ma/books?id=SedSMQAACAAJ.

[15] Szabo N. Formalizing and Securing Relationships on Public Networks. First Monday 2(9). doi:10.5210/fm.v2i9.548. URL: https://journals.uic.edu/ojs/index.php/fm/article/view/548..

[16] Labs P. IPFS Powers the Distributed Web, online; accessed 20 December 2019; 2019. URL: https://ipfs.io/..

[17] Azaria A, Ekblaw A, Vieira T, Lippman A, MedRec. Using Blockchain for Medical Data Access and Permission Management. In: 2016 2nd International Conference on Open and Big Data (OBD). Vienna, Austria: IEEE; 2016. p. 25–30. doi: https://doi.org/10.1109/OBD.2016.11.

[18] Liang X, Zhao J, Shetty S, Liu J, Li D. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE: Montreal, QC; 2017. pp. 1–5..

[19] Dagher GG, Mohler J, Milojkovic M, Marella PB. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain Cities Soc 2018;39:283–97. doi: https://doi.org/10.1016/j.scs.2018.02.014. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210670717310685.

[20] Dwivedi A, Srivastava G, Dhar S, Singh R. A Decentralized Privacy-Preserving Healthcare Blockchain for IoT. Sensors 2019;19(2):326. doi: https://doi.org/10.3390/s19020326. URL: http://www.mdpi.com/1424-8220/19/2/326.

[21] Hang Choi, Kim. A Novel EMR Integrity Management Based on a Medical Blockchain Platform in Hospital. Electronics 2019;8(4):467. doi: https://doi.org/10.3390/electronics8040467. URL: https://www.mdpi.com/2079-9292/8/4/467.

[22] Mamta BB, Gupta K-C, Li VCM, Leung KE, Psannis S Yamaguchi. Blockchain-Assisted Secure Fine-Grained Searchable Encryption for a Cloud-Based Healthcare Cyber-Physical System. IEEE/CAA J Autom Sin 2021;8 (12):1877–90. doi: https://doi.org/10.1109/JAS.2021.1004003. URL: https://ieeexplore.ieee.org/document/9416952/.

[23] Lu J, Shen J, Vijayakumar P, Gupta B. Blockchain-based Secure Data Storage Protocol for Sensors in the Industrial Internet of Things. IEEE Trans Ind Inf 2021:1. doi: https://doi.org/10.1109/TII.2021.3112601. URL: https://ieeexplore.ieee.org/document/9537295/.

[24] Dorri A, Kanhere SS, Jurdak R. Towards an Optimized BlockChain for IoT. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation – IoTDI '17. ACM Press: Pittsburgh, PA, USA; 2017. pp. 173–178. doi:10.1145/3054977.3055003. URL: http://dl.acm.org/citation.cfm?doid=3054977.3055003..

[25] Özylmaz KR, Yurdakul A. Integrating low-power IoT devices to a blockchain-based infrastructure: work-in-progress. In: Proceedings of the Thirteenth ACM International Conference on Embedded Software 2017 Companion – EMSOFT '17. ACM Press, Seoul, Republic of Korea; 2017. pp. 1–2. doi:10.1145/3125503.3125628. URL: http://dl.acm.org/citation.cfm?doid=3125503.3125628..

[26] Chakraborty S, Aich S, Kim H-C. A Secure Healthcare System Design Framework using Blockchain Technology. In 2019 21st International Conference on Advanced Communication Technology (ICACT), IEEE, PyeongChang Kwangwoon_Do, Korea (South); 2019. pp. 260–264. doi:10.23919/ICACT.2019.8701983. URL: https://ieeexplore.ieee.org/document/8701983/..

[27] Rifi N, Rachkidi E, Agoulmine N, Taher NC. Towards using blockchain technology for eHealth data access management. In: 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME). Beirut: IEEE; 2017. p. 1–4. doi: https://doi.org/10.1109/ICABME.2017.8167555.

[28] Sun J, Yao X, Wang S, Wu Y. Blockchain-based secure storage and access scheme for electronic medical records in IPFS. IEEE Access 2020;8:59389–401. doi: https://doi.org/10.1109/ACCESS.2020.2982964.

[29] Nguyen GN, Viet NHL, Elhoseny M, Shankar K, Gupta B, El-Latif AAA. Secure blockchain enabled Cyber-physical systems in healthcare using deep belief network with ResNet model. J Parallel Distrib Comput 2021;153:150–60. doi: https://doi.org/10.1016/j.jpdc.2021.03.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0743731521000642.

[30] Ethereum. Private Networks – Go Ethereum, online; accessed 19 December 2019; 2019. URL: https://geth.ethereum.org/docs/interface/private-network..

[31] Ethereum. Go Ethereum, online; accessed 19 December 2019; 2019. URL: https://geth.ethereum.org/..

[32] Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In: G. Goos, J. Hartmanis, J. van Leeuwen, K. Nyberg (Eds.), Advances in Cryptology – EUROCRYPT'98, vol. 1403, Springer, Berlin Heidelberg: Berlin, Heidelberg; 1998. pp. 127–144. doi:10.1007/BFb0054122..

[33] Qin Z, Xiong H, Wu S, Batamuliza J. A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing. IEEE Trans Services Comput 2016:1. doi: https://doi.org/10.1109/TSC.2016.2551238. URL: http://ieeexplore.ieee.org/document/7448446/.

[34] Azbeg K, Ouchetto O, Andaloussi SJ, Fetjah L, Sekkaki A. Blockchain and IoT for Security and Privacy: A Platform for Diabetes Self-management. In: 2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech). Brussels, Belgium: IEEE; 2018. p. 1–5. doi: https://doi.org/10.1109/CloudTech.2018.8713343.

[35] I.D. Federation., IDF DIABETES ATLAS Ninth edition 2019; 2019. URL: https://www.diabetesatlas.org/en/resources/..

[36] Piatkiewicz P. Hypoglycemia in Elderly Type 2 Diabetes Patients. Diabetes Management 2016:5.

[37] Web3j, web3j/web3j, online; accessed 20 December 2019; 2019. URL: https://github.com/web3j/web3j..

[38] Ethereum. ethereum/wiki, online; accessed 20 December 2019; 2019. URL: https://github.com/ethereum/wiki/Light-client-protocol..

[39] Ez DN. Umbral: a threshold proxy re-encryption scheme. NuCypher Inc and NICS Lab, University of Malaga, Spain; 2018. p. 8.

[40] Chaquo Ltd. The easiest way to use Python in your Android app, online; accessed 21 December 2019; 2019. URL: https://chaquo.com/chaquopy..

[41] Community I. ipfs/java-ipfs-http-client, online; accessed 21 December 2019 (Dec. 2019). URL: https://github.com/ipfs/java-ipfs-http-client..

[42] Remix. Remix – Ethereum IDE, online; accessed 14 June 2020; 2020. URL: https://remix.ethereum.org..

[43] Kumar R, Marchang N, Tripathi R. Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain. In: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS). Bengaluru, India: IEEE; 2020. p. 1–5. doi: https://doi.org/10.1109/COMSNETS48256.2020.9027313.

[44] Alamri B, Javed IT, Margaria T. A GDPR-Compliant Framework for IoT-Based Personal Health Records Using Blockchain. In: 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Paris, France: IEEE; 2021. p. 1–5. doi: https://doi.org/10.1109/NTMS49979.2021.9432661.

[45] Miyachi K, Mackey TK. hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design. Inf Process Manage 2021;58(3):. doi: https://doi.org/10.1016/j.ipm.2021.102535. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306457321000431102535.