Computer Communications 180 (2021) 31–47

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

BCHealth: A Novel Blockchain-based Privacy-Preserving Architecture for IoT Healthcare Applications

Koosha Mohammad Hossein [a], Mohammad Esmaeil Esmaeili [a], Tooska Dargahi [b,*], Ahmad Khonsari [a,c], Mauro Conti [d]

[a] University of Tehran, Tehran, Iran
[b] University of Salford, Manchester, UK
[c] School of Computer Science, Institute for Research in Fundamental Sciences, Tehran, Iran
[d] University of Padua, Italy

ARTICLE INFO

Keywords:
IoT
Healthcare
Blockchain
Privacy
Access control

ABSTRACT

The advancements in networking technologies have introduced the Internet of Everything (IoE) and smart living concepts. The main idea behind making everything smarter is to improve individuals' quality of life. An excellent example of such a technology is smart healthcare which provides efficient, sustainable, and real-time human services. However, data security and privacy are among the most important challenges of smart healthcare applications. Blockchain (BC) has been considered as a promising solution for the secure management of healthcare data due to its immutability and transparency features. However, there is a trade-off between transparency and user data privacy which is a prominent challenge in adopting BC for healthcare applications. Some researchers have considered user data privacy and proposed a few solutions; however, data owner's access control desire has not been considered in the state-of-the-art models. In this paper, to overcome the trade-off challenge between transparency and access control, we propose an architecture (so-called BCHealth) that enables data owners to define their desired access policies over their privacy-sensitive healthcare data. BCHealth is composed of two separate chains for storing access policies and data transactions. We address the real-world development challenges of BC, i.e., scalability, delay, and overhead, by adopting a clustering approach. Our extensive experimental analysis proves the efficiency of BCHealth (in terms of computation and processing time) and its resilience against several security attacks.

1. Introduction

Internet of Things (IoT) has been one of the top research areas in both industry and academia during the last few years. IoT refers to a network of objects (including sensors and actuators, embedded devices, smartphones, smart cars, etc.) connected via the Internet or wireless communications [1]. Ubiquitous information gathering and sharing across different platforms has led to the introduction of the Internet of Everything (IoE) and the application of IoT and IoE in smart living scenarios, such as smart healthcare and smart transportation [2]. Meanwhile, due to the rapid development of wearable sensors, networking technologies, and machine learning, healthcare became one of the main IoT applications. As reported by ZDNET [3], there is a worldwide interest in the usage of IoT devices for healthcare, such that the global spending on IoT for healthcare will reach $405.65 bn by 2026 [3].

In healthcare applications, IoE enables continuous remote monitoring of patients and improves the collection and analysis of health-related data [4]. In such a system, patients can be equipped with different sensors for real-time collection of their health-related data, such as heart rate, blood pressure, body temperature, and even their movements. The collected data by the sensors will (usually) be sent to a master device, e.g., a smartphone, which in turn sends those data to a database in a healthcare center (or a data center, or even cloud) for further analysis by medical staff or machine learning algorithms [5]. This will improve patients' experience in terms of quality of service and reduces their costs in terms of time and transportation, as their health condition is being monitored in real-time without a need for hospital visits [4].

Despite having many benefits, integrating IoE in healthcare systems brings about various challenges, out of which security and privacy of the sensed data are highly important [6]. Health-related data usually contains confidential and sensitive information about the patients

* Corresponding author.
E-mail addresses: kosha.hosseini@ut.ac.ir (K. Mohammad Hossein), me.esmaeili@ut.ac.ir (M.E. Esmaeili), t.dargahi@Salford.ac.uk (T. Dargahi), a_khonsari@ut.ac.ir (A. Khonsari), conti@math.unipd.it (M. Conti).

https://doi.org/10.1016/j.comcom.2021.08.011
Received 22 December 2020; Received in revised form 26 July 2021; Accepted 14 August 2021
Available online 24 August 2021

(e.g. patients' identities, health status, and even visual data such as medical images), where the leakage or tampering of this data could endanger the patient's life. In some articles, e.g., [7–9], have been shown that sensitive and meaningful information could be extracted from visual data. This data will be sent over a possibly untrusted network and stored in (most probably) centralized servers of a healthcare center, which will open up the doors for several attacks and inefficiencies, including single point of attack/failure [10].

On the one hand, centralized storage of such data has been proven to cause several issues, such as [2]: (i) servers might be prone to Denial of Service (DoS) attack or failure (due to communication and computation overhead), (ii) patient's data might be accessed by unauthorized parties or tampered by authorized users (possibly unintentionally), and (iii) patients would need to trust a third party for storing and processing their personal data which is not suitable in a distributed environment of large-scale healthcare scenarios [2,11,12]. On the other hand, traditional methods for preserving security and privacy are not efficient in healthcare scenarios as: (i) resource-constrained IoT devices cannot afford computation and memory-intensive operations, and (ii) approaches that add noise to the collected data (to preserve privacy) will diminish the usability of data in healthcare applications [13]. Therefore, there should be proper access control mechanisms to protect the confidentiality of the data while preserving their usability [14].

To address these issues, recently, several researchers proposed the usage of BlockChain (BC) technology in healthcare systems [15–20]. BC is a cryptographically secured immutable timestamped public ledger used to store and share data in a distributed manner (through peer-to-peer communication) [21]. The BC network comprises several nodes (i.e., computing machines) that validate each requested transaction and store information of the verified transaction [22]. It has been proven that BC provides security, efficiency, and transparency when used in information exchange scenarios [22]. Due to its outstanding characteristics, BC is a promising solution for data exchange and storage in IoT and healthcare systems. Its decentralized feature eliminates the need for the patients and hospital management system to trust the third party or a central data storage/authority and addresses the single-point-of-failure/attack issue [23,24]. BC's immutability and irreversibility features ensure that the patient's data has not been removed or maliciously tampered with. Moreover, the anonymity feature, which is provided by the usage of pseudonyms instead of the users' real identifiers, preserves the patients' privacy [25].

Despite having many advantages, BC has various issues, such as scalability, energy consumption, delay, and memory overhead imposed on the participating nodes in a large-scale network [22]. These issues are even more challenging when adopting BC in IoT applications with many devices involved. The vast amount of exchanged and stored data in IoT scenarios imposes high network and memory overhead and delay in the data processing. In particular, in the case of non-delay-tolerant healthcare systems with memory and computational resource limitation, the usage of BC is not straightforward [26]. To tackle these issues, researchers have proposed some solutions for adopting BC in IoT. Dorri et al. [27] proposed a lightweight architecture in which a distributed trust model has been used to reduce the processing time of block validation in smart home scenarios. Moreover, the usage of clustering to decrease the network overhead and delay has been proposed in a few research papers, such as [26,28].

To the best of our knowledge, the patients' ability to define access policies over their sensitive data remains an under-investigated issue. Patients need to trust the healthcare center and medical staff to treat their data responsibly, while they do not have any control over their own sensitive data. Also, in the existing BC-based schemes in the literature, patients do not have control over where their data has been stored and who can access them. This motivated us to propose a new BC-based access control architecture (called BCHealth) for storing and retrieving healthcare records, which is user-centric, fully distributed, and practical. The main aim is to remove the need for a trusted third party and instead give full control over their healthcare data to patients through a user-controlled IoT Healthcare Manager (IHM). This will help in preserving the privacy of users, while also facilitating the usage of advancing distributed machine learning techniques, such as federated learning [29]. Moreover, to address the common scalability and performance challenges of blockchain, we take advantage of node clustering and also using two separate chains for storing the user's data and access policies. BCHealth is a thorough extension of our previous study [30].

Contribution. This paper brings the following contributions:

(1) BCHealth allows data owners to define access policies over their sensitive data. In BCHealth architecture, the user's data is not published on the BC network without the user's permission (unlike most BC-based related work in which the data is published on third-party servers and then access control policies are applied). BCHealth stores the data locally on a machine, which is nearest to the data owner (similar to the edge computing concept) instead of a cloud or healthcare data center. This will significantly reduce the delay and communication overhead.
(2) BCHealth introduces the usage of two different chains, namely, data chain and access control (policy) chain. Data chain stores patient's healthcare data and access control chain stores the patient's defined access policies in a private BC. It preserves the confidentiality of the data by storing the hash of patient's data as transactions in the data chain. At the same time, to control access over data, patients store their desired access policy in another chain. This ensures that the data owner's access policies will remain unaltered, and access to patient's data will be controlled as expected.
(3) BCHealth uses a new clustering approach to increase the BC network throughput and improve the scalability of the network. Here, instead of considering a Cluster Head (CH) for each cluster, we introduce a hierarchical structure. We allocate the first two bytes of the data packets to the cluster number associated with that data. Upon receiving a data packet, each cluster member will be able to identify the cluster that this data belongs to. This new approach reduces the unnecessary reliance on a CH which imposes delay and risk of a single point of failure.
(4) BCHealth is equipped with an alarm system for emergency situations, which will notify corresponding medical staff if urgent action is required based on the patient's health condition. For example, in the case of the COVID pandemic, BCHealth could help in informing the medical staff as soon as identifying the disease symptoms for them to take appropriate actions.
(5) We use a new modified BC network and Proof-of-Authority (PoA) consensus algorithm to improve the performance and scalability of the system. We evaluate the performance of BCHealth by implementing its components in Python. Through several experiments, we show the efficiency of BCHealth in terms of improved scalability and delay. We also discuss its resilience against different security attacks.

The rest of the paper is organized as follows. Section 2 explains the required background on blockchain and explores the related work. Section 3 presents the system model and briefly introduces the different entities of the proposed architecture, while Section 4 describes the proposed architecture in detail. Section 5 evaluates the BCHealth from the security and privacy perspective, while Section 6 provides the experimental analysis results. Section 7 compares the BCHealth with the state-of-the-art, highlights our contribution and discusses the performance of our proposed architecture, and Section 8 concludes the paper and proposes future work directions.

2. Background and related work

This section provides the required background on blockchain architecture, the mining process, and consensus algorithms. We then review the state-of-the-art and discuss the advantages of BCHealth compared to related work.

2.1. Blockchain (BC)

BC could be considered as a distributed database of committed transactions in a network. It is actually a peer-to-peer network of several participating nodes. BC uses a structure called *"block"* to record all the transactions that are sent from different nodes in the network [31]. Each block has a pointer (i.e., cryptographic hash) to its previous block, which will lead to the formation of a *"chain of blocks"*. This chain structure provides the immutability feature such that any small change to each block will break the chain structure and reflect a malicious/erroneous transaction in the network [2]. In each transaction submitted by a user (i.e., the sender), the addresses of the sender and the receiver are specified by their public keys (PK) instead of their identities. This method provides anonymity for both parties [32].

2.1.1. Miners and mining

Upon receiving a transaction, each node in the BC network will broadcast the transaction to other nodes in the network. Some specific nodes in the BC network are known as *"miners"*, which are responsible for providing security of the BC, verifying each transaction, and generating blocks. In order to add this new block to the chain, each miner needs to solve a compute-intensive mathematical puzzle. Once a solution is found, the corresponding miner can generate a new block and send it to the other miners in the network. The last step of adding a block to the chain is verification of the generated block through running a "*consensus algorithm"* among the miners. If the majority of the miners agree on the correctness of the new block, this block will be added to the chain. This whole process of solving the problem and adding the new block to the blockchain is called *"mining"* [2].

2.1.2. Consensus algorithms

Most of the existing BC implementations use one of the following consensus algorithms: Proof of Work (PoW) [33], Proof of Stake (PoS) [34], or Proof of Authority (PoA) [35]. However, POW demands high computational resources, while PoS needs both computational and memory resources to solve the cryptographic puzzle [23]. Furthermore, due to the complexity of solving the cryptographic puzzle (e.g., Bitcoin), validation and confirmation of transactions in PoW consensus algorithms require more time, which reduces the number of transactions that can be recorded in BC. In the case of an IoT network, in which (usually) a large number of transactions are generated by a large number of devices per second, PoW or other compute-intense consensus algorithms are not appropriate.

An alternative solution (which we use in BCHealth) is Proof-of-Authority (PoA) algorithm [35]. PoA is a new family of Byzantine Fault-Tolerant (BFT) consensus algorithms that operate on permissioned blockchain. PoA improves performance due to lighter message exchange, compared to the traditional Practical Byzantine Fault Tolerance (PBFT) [35]. In the PoA algorithm, there are N nodes in the network, which are called authorities, where each authority is specified with a unique identifier, assuming the majority $(N/2 + 1)$ of them are honest [35]. The consensus process in the PoA algorithm is based on a rotational mining algorithm. In this algorithm, time is divided into some slots, and at each slot, an authority is selected as the mining leader. The POA was originally designed to be used in the private Ethereum network and implemented in both Aura and Clique versions [35]. PoA is used by Parity [36], and Geth [37], two well-known clients of Ethereum. Researchers [38] reported that PoA is a promising algorithm to be used in the development and maintenance of distributed applications due to its features: (1) it requires less computational power compared to PoW; (2) it has a higher transaction rate due to the generation of blocks at specific time intervals by authorized nodes (this increases the speed of validating transactions in BC network); (3) it is more scalable compared to PoW. Therefore, in BCHealth, we adopt the PoA consensus algorithm in our BC Network.

2.2. Related work

In this section, we review literature related to the usage of blockchain in healthcare scenarios (Section 2.2.1), and related work that introduce the usage of clustering in blockchain-based models (Section 2.2.2). In Table 1 we provide a comparison between BCHealth and the existing literature on storage and retrieval of healthcare data. Table 2 compares BCHealth with the state-of-the-art blockchain-based healthcare systems from different perspectives, including scalability, energy consumption, adopted consensus algorithm, throughput, usage of clustering, single point of failure prevention and node recovery consideration.

2.2.1. Healthcare

Recently the usage of the BC network in the healthcare domain has attracted the interest of many researchers [17,18,46,47]. In [15] a system for handling Electronic Health Records (EHRs) is proposed which uses BC to preserve the privacy of medical information. This system provides patients with an immutable and comprehensive log of their data and enables easy access to their medical information across providers and treatment sites. In fact, the authors use BC for medical data access and permission management. In [48], authors propose a blockchain-based method to eliminate the risk of a single point of failure and divide the high computational load of Attribute-based searchable encryption (ABSE) between blockchain nodes in a cloud-based storage and retrieval method to protect the privacy of health data. In [49], authors have proposed a method for auditing data on the cloud-based on the Merkle Tree. The paper [50] proposes secure invasion discovery with a blockchain-based data communication with a classification model for Cyber–physical systems in the healthcare area. Other researchers in [16] propose a user-centric solution for health data sharing, which protects the privacy of data by using a decentralized and permissioned BC. The work [51] aims to create a secure Cache Decision Method in a wireless network operating over a Smart Building. In this research, data is collected from medical devices and is made available to the medical staff via the cloud. BC is used for maintaining the integrity of the stored data in the cloud. In [26] a new clustering model based on BC for healthcare has been proposed. This work does not use the PoW consensus algorithm to be more efficient for IoT. Instead, it adopts ring signature to provide anonymity for users and considers double encryption of data by a lightweight encryption algorithm (ARX ciphers) to guarantee users' security and privacy.

2.2.2. Clustering

As mentioned earlier, adopting BC technology in IoT networks has several challenges. One challenge is related to the miner nodes as all of them need to store the latest version of the replicated data. Although this is a good feature for systems that require higher levels of trust, e.g., financial systems, it is not cost-effective for a large number of transactions in IoT networks. Other challenges include high network overhead due to consensus operations (especially in PoWs) and low throughput due to the number of transactions that can be recorded in BC [23].

Almost all of the research studies on the usage of clustering in the literature follow a similar approach. They consider a central node as the CH, which manages the whole cluster and adds new blocks to the cluster. Moreover, CHs are responsible for accessing the user data, i.e., the data access request is sent to the nearest CH. If the requested data does not exist in that cluster, the request will be sent to the

Table 1
A comparison between the existing methods of storing and retrieving medical records and BCHealth.

Research	Method	Advantages and Disadvantages
[39] (2021) [40](2021)	Central storage of the users' health data on the cloud and servers of medical centers is suggested.	Advantage: Easier data management and less delay in accessing the data. Disadvantage: risk of a single point of failure and attack, reliance on a third party for handling sensitive data, lack of user control over the data and risk of a privacy breach.
[41](2020)	Usage of blockchain for storing data.	Advantage: The challenges of centralized storage have been resolved. Disadvantage: performance is reduced due to delays and imposed storage space overhead in all the blockchain nodes (as they all store a copy of the data). Moreover, users still do not have control of their data, and the data cannot be deleted.
[42] (2018) [43](2019)	Storage of the data summary (hash of data records) on the blockchain rather than the whole data record is suggested. The healthcare data will be stored on the servers of medical centers.	Advantage: data manipulation can be prevented; delay and memory usage have been improved. Disadvantage: data is still stored centrally and challenges of central storage of data exist.
[44] (2020) [45](2020)	Blockchain is used as a storage node manager, and health data is stored in a distributed environment such as IPFS (InterPlanetary File System).	Advantage: reduced network delay, decentralized storage, and data integrity. Disadvantage: users still do not have control of their data.
BCHealth	Blockchain is used as a decentralized controller to store the hash of the data. An extra chain is used to store the user access policies. Users actual data are stored in a distributed manner, on devices closer to the user and controlled by the user. The clustering approach is used to improve the performance of blockchain.	Advantage: the benefits of storing the hash of the data on the blockchain (such as immutability, distribution and security) are retained. Compared to the other BC-based approaches, the delay and storage usage is decreased due to the usage of clustering. This architecture is user-centric, and patients have complete control over their data, they can prevent the spread of their data and define access control policies over their data. This architecture is compatible with the usage of distributed machine learning algorithms for further improvement and distribution of user data analysis. Disadvantage: compared to the centralized approaches the delay in accessing the health records and storage overhead has been increased.

next CH, and this process will be continued until the right cluster is found [52]. However, this approach of clustering and searching for data has two issues: (1) the CH might become a single point of failure. For example, if one CH is not accessible by other nodes/CHs (due to an attack or malfunctioning), the whole cluster will be inaccessible; (2) the data access process is time-consuming and imposes delay due to the need to search different clusters one by one to find the right cluster.

Compared to the state-of-the-art models in BCHealth, we group miners into several clusters in a hierarchical structure. Each cluster contains some miners of the whole BC network. In addition, a special cluster will be assigned to each user for storing his transactions. The hierarchical structure will be integrated into user IDs to reduce the delay in searching the right cluster. This approach will also reduce the required storage space in each miner, facilitate network management, speed up data access process, and reduce the length of the chain in each miner. Moreover, to the best of our knowledge, controlling the access to patient data while using BC has remained uninvestigated in the literature. Researchers in [53] explained that the irreversibility nature of BC and the fact that a copy of the ledger is stored in each node would make it difficult to use BC for privacy purposes. Although some research studies, e.g., [54], present the application of BC in the privacy protection of personal data. We address the issue of access control in BC by providing the data owners with a means of defining their desired access policies in BCHealth. Details of our proposed architecture are provided in Section 4.

In Table 2 we compare BCHealth with recent related work on blockchain-based healthcare data management; this includes those methods that use clustering as well. In this table, the "clustering" column specifies whether a clustering method is used. The "scalability" column specifies whether improving the scalability of BC has been considered in the paper. The "energy" column indicates whether the proposed method is energy efficient. The "Throughput" column determines whether the proposed method has improved throughput. The "SPOF (Single Point Of Failure)" column indicates whether the method prevents the single point of failure or not. It is worth mentioning that "single point of failure" is not an issue in the original blockchain design. However, some methods have considered reducing the number of nodes involved in consensus operations to improve scalability and energy consumption. For example, cluster management is assigned to a specific node (CH) in some of the proposed clustering methods, which might become a single point of failure/attack for that cluster. The "node recovery" column refers to those nodes that are compromised or damaged. Similar to the previous column, in the original blockchain system, this problem does not exist due to the fact that many nodes are involved; however, in clustering-based methods, a mechanism must be in place to identify the damaged nodes and replace them.

3. System model

In this section, we explain different entities and concepts that are involved in our proposed architecture. BCHealth is composed of several entities: (1) Wearable sensors and patient's smartphone; (2) IoT Health Manager (IHM) for receiving data from user's device and storing a hash of them in the BC; (3) Blockchain (BC) to store user's data hash and access policies; (4) Miners; (5) Healthcare servers that are responsible for creating primary permissioned Blockchain and managing the clusters; and (6) Healthcare wallet associated to each user for storing his personal information, such as private keys, access policies, etc. We explain the details of each entity in the rest of this section.

3.1. Wearable sensors and smartphone

In our system, each user is equipped with several wearable sensors, which gather data about the user's health situation. These sensors are resource-constrained, and most of them (especially those implanted inside the patient's body) cannot perform computation-intense processes. Therefore, we consider the user's smartphone as a master device, which collects the data gathered by the sensors and stores them. The smartphone is responsible for encrypting and classifying the data based on their type (i.e., type of the sensor that collected them) and sending the data to a more powerful device, the so-called IoT Healthcare Manager (IHM). The smartphone uses a symmetric key encryption method to encrypt messages and send them to the IHM. An asymmetric encryption method, such as RSA or Diffie–Hellman, could be used to exchange keys between the user's smartphone and IHM. The user's smartphone is responsible for the following two main tasks.

Table 2
Comparison between the blockchain-based healthcare systems and BCHealth.

Research	Clustering	Scalability	Energy	Consensus	Throughput	SPOF Prevention	Node Recovery	Remarks
[24](2016), [55](2018)	✘	✘	✘	PoW	N/A	✓	✓	Scalability and high energy consumption due to the usage of PoW consensus algorithm are the main challenges.
[47](2018)	✘	✓	✘	PoW	N/A	✓	✘	Energy consumption is a challenge; however scalability is improved due to the usage of fewer nodes for consensus operations and only storing the hash of the data; damaged nodes could not be replaced.
[56]	✘	✓	✓	BPFT	N/A	✓	N/A	Due to the usage of BPFT, this method is both scalable and energy efficient.
[57]	✘	N/A	✘	PoW	N/A	✓	—	Scalability and high energy consumption due to the usage of PoW consensus algorithm are the main challenges.
[58]	✓	✓	✓	BPFT	✓	Reduced but not solved	✘	Clustering and BPFT consensus algorithm are used to improve scalability and energy efficiency. A blockchain manager (BCM) is used for each cluster that maintains a copy of the ledger. No solution is proposed for compromised or damaged BCM and nodes in each cluster.
[59] , [60]	✓	✓	✓	PoA	✓	✘	✘	Clustering and PoA consensus algorithm are used to improve scalability and energy efficiency. The article uses CH for cluster management. No solution is proposed for compromised or damaged CH and nodes in each cluster.
BCHealth	✓	✓	✓	PoA	✓	✓	✓	We used clustering and PoA consensus algorithm to improve scalability and energy efficiency. BCHealth does not rely on CH, and due to the usage of the PoA mechanism, each node in the cluster is able to cross-manage the cluster nodes. We propose a solution to tackle the issue of a damaged node.

- Play the role of a gateway for sensors; categorize the data and send them to the IHM over the network.
- Display analysis report of the user's health status by contacting the IHM.

3.2. IoT Healthcare Manager (IHM)

The patient's healthcare data must be continuously monitored, stored, and analyzed. This requires high computational, energy, and storage resources. One potential solution is the usage of Cloud data centers, which requires the users to trust a third party (i.e., Cloud service provider). Moreover, storing data on the Cloud will impose delays in data transfer and processing. Therefore, Cloud usage is not the best solution for applications requiring real-time data processing, especially healthcare applications. A better approach, which has attracted much attention recently, is Edge Computing (EC). It brings storage and processing resources to the proximity of the end-user [61]. EC focuses on reducing the communication overhead and delay by distributing the data processing and storage to the edge servers/devices [51].

In the IoT ecosystem with a large number of smart devices, some devices, which are called IoT management devices, can be used to collect and process data [62]. Considering the EC concept and its advantages, we present IoT Healthcare Manager (IHM) in our BCHealth architecture. IHM can be a computer with sufficient resources, which plays the role of a black box for each patient. We included the IHM in our system model with the assumption that in the future, every human being would have a system that logs all the different information about his/her state of health and also can analyze this data [63]. IHM is an intermediate entity between the user's smartphone and the BC network. It receives data from the patient's smartphone, stores them, and computes a hash of the data. It then sends the hash as a transaction to the BC using Health Wallet (HW), an application similar to cryptocurrency wallet (refer to Section 3.6). IHM is able to perform different kinds of data analytics, such as diagnostic, predictive, and prescriptive analytic for smart health monitoring [64]. For example, a patient's emergency status can be predicted by IHM to alert medical staff in advance. Usage of IHM has three main advantages:

1. It eliminates the need for storing data on a third-party server, which helps to provide better privacy and security protection of sensitive healthcare data;

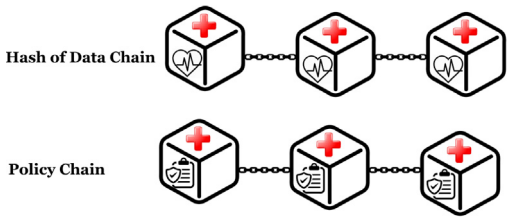

Fig. 1. Policy and data chain structure in BCHealth.

2. It performs several types of data analytics, which reduces the computational load on sensors and smartphones leading to reduced energy consumption;
3. Using IHM, compared to Cloud, brings the heavy AI processing operations close to the end-user, which reduces network latency. The results of data analysis will be accessible to the data owner through the application installed on his/her smartphone. This data is available only for authorized staff or due to critical health conditions.

3.3. Blockchain (BC)

BCHealth consists of a network of healthcare centers that play the role of participating nodes in the BC network. We take advantage of consortium BC [65] due to the variety of independent healthcare centers that require to be involved in our BC networks. The main feature of consortium BC is its partial immutability and partial distribution. Due to these features, it is not as strong as public BC, but it is more flexible and more scalable in terms of latency and storage requirements in public BC [66].

We consider two separate chains in our BC network for two types of transactions: (1) Data Chain: for storing transactions related to healthcare data, and (2) Policy Chain: for storing transactions that specify the access control policy defined by the data owner. Fig. 1 shows the structure of these two chains. These two chains are generally used for two reasons: (i) to keep the size of the policy and data chains smaller, which leads to an accelerated search operation in each chain (in particular the access control process), and (ii) due to the structural differences between the policy and data transactions, storing them in two different chains improves the BC management.

3.4. Miners and mining

In BCHealth, each healthcare center is associated with several miners, which are responsible for processing transactions, authenticating users, and verifying user's access to healthcare data (refer to Fig. 2). These miners could be distributed in different locations (i.e., healthcare centers) of the city. In addition, every health center must have a few nodes as miners (or authorities), which are authorized by the Ministry of Health (of each country) to serve as a miner in the BC network. These nodes are then divided into clusters and are responsible for performing transaction validation operations, similar to the Sharding method [67] in a traditional database. Thus, each user/patient is assigned a specific cluster upon registering to the healthcare system. Each user belongs to a specific cluster, while one cluster will be associated with several users.

As mentioned earlier, we adopt private BC to improve scalability, flexibility, latency, and network throughput. However, the main challenge of using private BC in our architecture is its immutability feature, as it makes it much easier for a group of compromised participants to change the data stored in the BC. To address this issue, instead of allocating one cluster to the miners of one healthcare center, we allow the miners in each cluster to be allocated to different healthcare centers. This reduces the chance of compromising a group of participants from

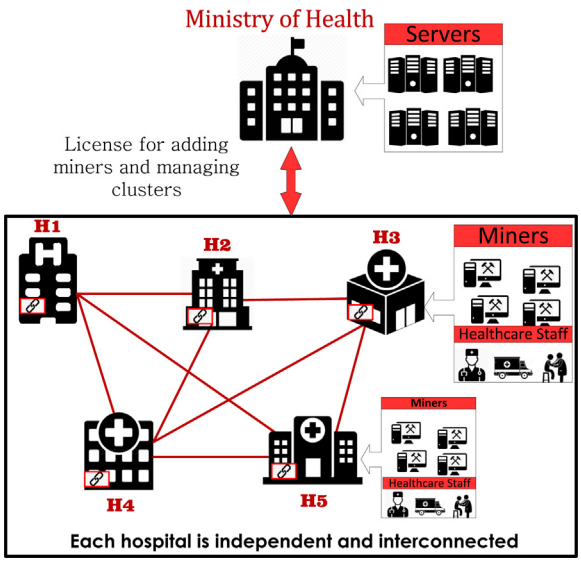

Fig. 2. Miners, healthcare centers and ministry of health in our system model.

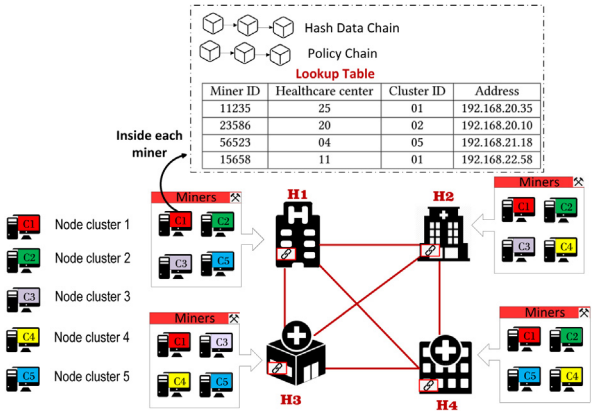

Fig. 3. A BCHealth architecture comprising four healthcare centers and five different clusters.

the same cluster by an attacker. Moreover, increasing the number of miners in each cluster could improve the network's immutability due to the reduced chance of compromising miners from the same cluster. However, this allocation decreases the degree of parallelism in the network.

Fig. 3 depicts a BCHealth architecture comprising four healthcare centers and five different clusters. Nodes of each cluster may belong to different healthcare centers. To be specific, the BC and miners in BCHealth architecture fulfill the following tasks:

1. Miners control and monitor any attempts to access the users' data based on the stored users' access policies.
2. Miners manage decentralized and peer-to-peer communication between patients and healthcare staff. This requires each cluster to maintain and manage the IHM address of its users.
3. Miners detect modification of data stored in the IHM of the user. Any modifications on the data values in the IHM produces a different hash, which does not match with the original hash of the block. This notifies the authorities that the original data has been tampered with.

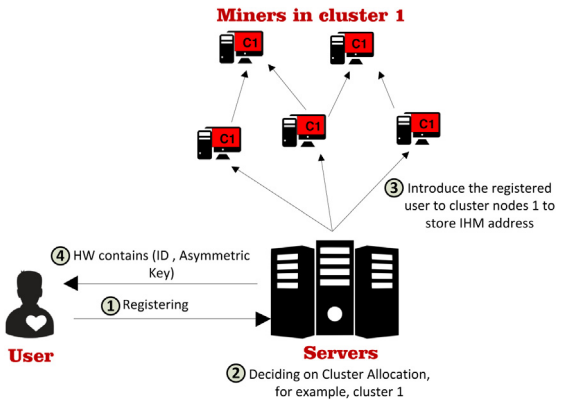

Fig. 4. Users registration process.

3.5. Healthcare servers

In our architecture, we assume that the Ministry of Health in each country has several servers that will be used for user and miner registration, cluster management, such as adding and removing nodes, and fault tolerance. We consider the servers as trusted and reliable, such that they operate under the supervision of the country's Ministry of Health. It should be noted that these servers do not store patients' data.

User Registration: Users (i.e., patients and medical staff) who want to use this system should first register and receive an ID and a key pair (public/private) from the healthcare server. The server will associate each user to a cluster (where his health policy and a hash of data will be stored). The cluster assignment decision is made based on several conditions, such as the amount of load on each cluster. The first two bytes of the user's ID will be used to store their associated cluster number. In BCHealth, we use each user's ID instead of his public key. User ID is generated as:

$$ID = Cluster\_Number \| Hash(PK \| Nonce) \quad (1)$$

where $Cluster\_Number$ is two-byte and specifies the user's cluster number. For example, if $Cluster\_Number$ is "01", it means that the user's transaction should be stored in the cluster with id "01". PK is the user's public key, $Nonce$ is a random number, and $Hash$ is a hash function, e.g., SHA256, as it is used in the rest of the paper.

The process of registering users is illustrated in Fig. 4 and consists of four steps as follow:

1. The user sends a registration request to the server.
2. In the second step, the server assigns a cluster number to the user. In this figure, we assume that cluster number one is assigned to the user.
3. Then, the server must introduce the newly registered user to the miners of the corresponding cluster. To this end, it generates a transaction and transmits it to the network of miners in this cluster. This transaction contains the user's public key and user ID. In our example, the first two bytes of the ID start with 01 (depicting the cluster number), and the rest is generated based on Eq. (1). Moreover, the user's IHM address will be stored in all the nodes of this cluster.
4. In the last step, the user will receive his ID along with a pair of public and private keys to be stored in his wallet (refer to Section 3.6).

Miner node registration and cluster management: Healthcare servers are responsible for the initial formation of clusters, including authentication, addition, and miner node removal. To this end, each healthcare center in a country is required to introduce a specific number of nodes as its miners. These nodes must be registered and authenticated by the servers of the healthcare center and receive a unique ID. Just after this phase, the miner nodes will be able to participate in the block validation process. Using this method, we generate a permissioned blockchain in which only permitted nodes (by healthcare servers) can join the network. After registering the miners in the central server, the server divides them into clusters based on different criteria, such as geographical diversity, to protect against cluster failure due to natural disasters or node compromise attacks within each cluster. Table 3 shows an example of a lookup table stored in each miner node as well as healthcare servers. It includes the ID and address of the miners and the information of the cluster assigned to each miner node by the healthcare center. In addition, the healthcare servers store an updated list of miners in each cluster and a list of users (patients/staff) in each cluster with their IDs. The list of users is also distributed to the miners in each cluster through a secure channel.

Table 3
An example of the lookup table that is stored in each miner node.

Miner ID	Healthcare center	Cluster ID	Address
11235	25	01	192.168.20.35
23586	20	02	192.168.20.10
56523	04	05	192.168.21.18
15658	11	01	192.168.22.58

Table 4
Notation Table.

Parameters	Description
U_i	Users registered in the system, $i \in \{1 \dots m\}$
T_j	Data or policy transaction
H_i	Healthcare centers or hospitals, $i \in \{1 \dots n\}$
M_i	Miner node i, where $i \in \{1 \dots q\}$
C_i	Cluster number i, where $i \in \{1 \dots k\}$
$\mathcal{P}$	Doctor and medical staff
SP_{U_i}	User i's Smartphone
S_i	Sensors, $i \in \{1 \dots z\}$
D_{U_i}	User i's health data
$hash_D$	Hash of data

3.6. Healthcare Wallet

Each registered user in the system is assigned an application similar to a cryptocurrency wallet, which we call Health Wallet (HW). All the registration information, which is explained in Section 3.5, will be recorded in the user's HW. The HW stores the user's ID, as well as public and private keys. Each user will define and manage his access control policies and account data inside the HW. Moreover, required information regarding the cluster that is assigned to the user will be stored in his HW.

4. BCHealth architecture

This section provides a detailed explanation of BCHealth system functionality. The complete architecture is demonstrated in Fig. 5. To make it easier to follow the whole procedure of generating, storing, and accessing users' health data. We consider a running example. Table 4 lists the notations that are used in the rest of this section.

We assume a set of m users $\{U_1 \dots U_m\}$ and a set of n healthcare center $\{H_1 \dots H_n\}$ who are interested in joining the BCHealth architecture. Each user U_i has his own IHM which is presented with IHM_{U_i}. Each hospital H_j has a set of q miners $\{M_1 \dots M_q\} \in H_j$ that we present with M_{H_j}. For example, M_{H_1} is a miner in hospital H_1. We also consider a set of k clusters in our architecture, i.e., $\{C_1 \dots C_k\}$. Each miner is associated with a cluster, e.g., $M_q \in C_p$, which means that miner M_q is associated to the cluster C_p. On the other hand, each cluster C_p is composed of l miner nodes, each of which belonging to a possibly different healthcare

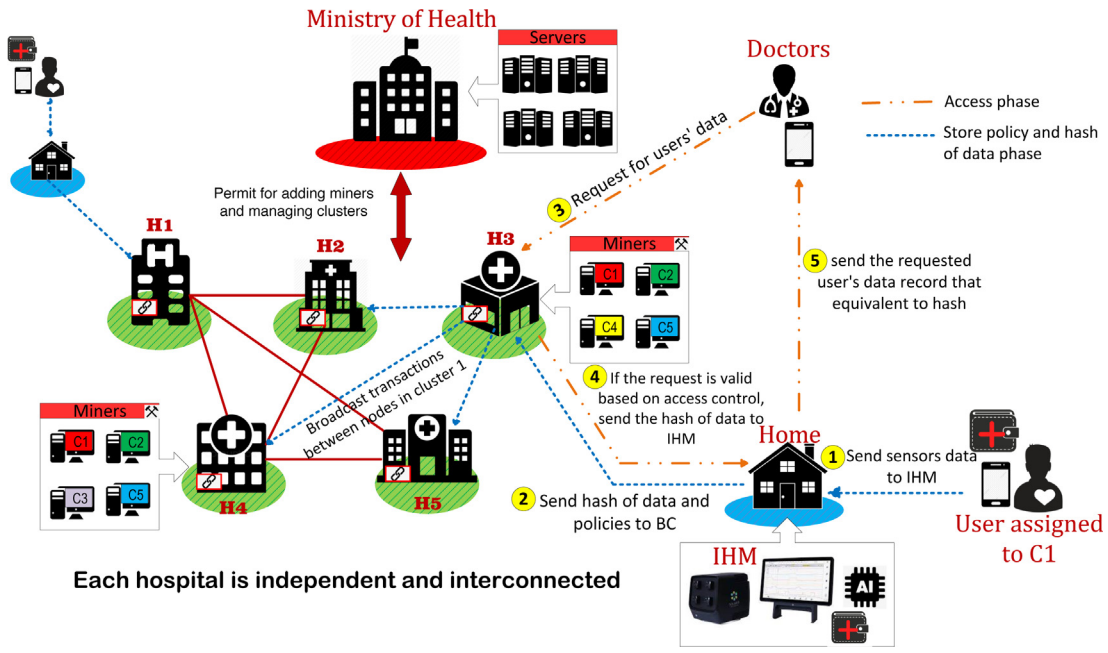

Fig. 5. Proposed blockchain-based healthcare architecture (BCHealth).

center, e.g., $\{M_{H_1} ... M_{H_n}\} \in C_p$. For the sake of simplicity, we use $M_q C_p$ notation to indicate that the miner q is a member of the cluster P.

In what follows, we consider user U_1 and explain the required steps for storing his healthcare data in BC, which will be requested and accessed by medical staff $\mathcal{P}$, if he is authorized. Fig. 6 presents a high-level overview of these steps.

1. User U_1 is registered with hospital H_x and is associated to cluster C_1. She is equipped with a number of sensors $\{S_1 \dots S_z\}$ that collect her health data D_{U_1}. This data is sent to the user's smartphone, SP_{U_1}, for encryption, classification, and sending to the IHM. The SP_{U_1} encrypts the data using a symmetric key algorithm (the key has already been shared between these two devices via an asymmetric key algorithm) and sends it to the IHM_{U_1}, i.e., $SP_{U_1}(Enc(D_{U_1})) \rightarrow IHM_{U_1}$.
2. The IHM_{U_1} receives the encrypted data, decrypts it, and stores the data and hash of the data ($Hash_D$). Then IHM_{U_1} generates a transaction T_i containing $Hash_D$, encrypts and signs the hash (i.e., $Sign_{U_1}$) and sends it to one of the miners (chosen randomly or the nearest miner) in the hospital H_x, i.e., $IHM_{U_1}(T_i) \rightarrow (H_x M_y)$ When a miner receives a transaction, it sends an acknowledge (ACK) message to the IHM. If, after a threshold period, the IHM does not receive any ACK message from the target miner, it will send the transaction to another miner.
3. The miner M_y checks the first two bytes of the user ID in T_i to figure out the cluster number that U_1 belongs to (i.e., C_1). Then, it broadcasts T_i to all the miners of C_1. This means, T_i will be stored in all the miners of this cluster, i.e $M_y(T_i) \rightarrow \{M_1C_1, M_2C_1, \dots, M_lC_1\}$. Each of these l miner nodes validate the transaction, e.g., $M_lC_1 \rightarrow check(ValidateTransaction(T_i))$, and eventually store it in the pre-defined block format.
4. At this stage, assuming that the medical staff $\mathcal{P}$ wants to access U_1's data, he should generate a transaction T_j containing U_1's ID, type of sensors, and the data access range. Then she transmits it to one of the hospital's miners randomly, that is $P(T_j) \rightarrow random(M_qC_n)$.
5. The miner node M_q checks the U_1's ID in the T_j transaction to identify the cluster number that U_1 belongs to. If the user is associated to the same cluster as M_q's cluster, then M_q validates the transaction and checks the access control policy for $\mathcal{P}$ inside the "policy chain" (note that in the third step, policies and a hash of the data chains are mined and verified by all of the miners in the same cluster, therefore all miners in the same cluster have stored the same blocks and transactions). If the access policy is valid and $\mathcal{P}$ are allowed to have access to U_1's data, then M_q extracts the $hash_D$ from the "data chain". Then generates a new transaction containing a hash of the requested data, $hash_D$, and sends it to the IHM_{U_1} to request for the original data. For this step, we assume that every miner node knows the IHM address of the users that are associated with its cluster; this information exchange has been performed during the user registration phase explained in Section 3.5.
6. After the validation of the new transaction by IHM_{U_1}, the equivalent plain data will be found in the database inside the IHM and sent to the $\mathcal{P}$'s address after encrypting with the public key of $\mathcal{P}$.

4.1. Data owner access policy definition

In BCHealth, the data owner defines explicit access policies through HW (explained in Section 3.6). The data owner stores the desired access policies in BC as a special type of transaction, call *"Policy Transaction"*, which are stored in the policy chain of the BC network. To this end, the data owner can look up in her registered health center's directory to find the medical staff's ID and grant access to him. This will not only provide immutability for access policies but also speeds-up the search operation in each chain. Obviously, each user can define and apply one policy at a time for a collection of data. For example, Alice can define a policy transaction that a physician can access all of her health data or specify access to data within a specific time interval (e.g., two months). Each policy can be specified as a 7-tuple, as follows:

$$\langle ID_{\text{owner}}, ID_{\text{req}}, Type, T_e, \langle D_s, D_e \rangle, \text{ Valid } \rangle$$

- ID_{owner} : The ID of the data owner, i.e., the patient, for example Alice.
- ID_{req} : The ID of the person who can access the data, e.g., Dr. Bob.
- $Type$: The type of data that the authorized users can access, e.g., ECG data.
- T_e : The expiration date of policy.

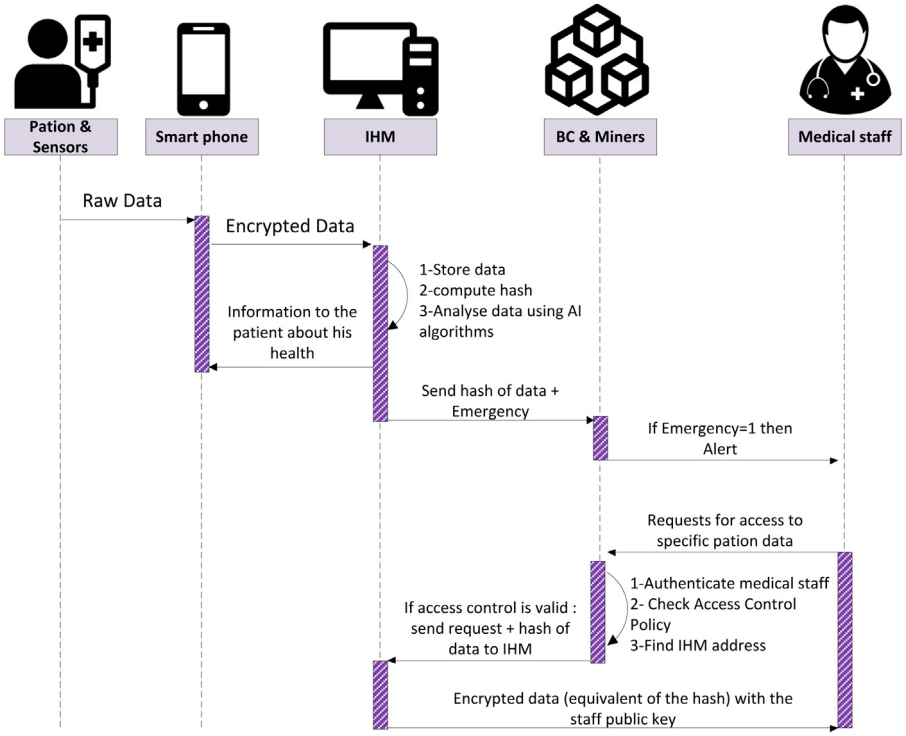

Fig. 6. Overview of exchanges between different component in one overview.

Table 5
Two example policies.

ID_{own}	ID_{Acc}	$Type$	T_e	D_s	D_e	$Valid$
A	X	Heartbeat	20/11/2020	20/05/2016	20/05/2018	1
A	Y	ALL	09/05/2020	07/06/2018	07/07/2018	0

- $\langle D_s, D_e \rangle$: This attribute specifies the period of data availability. For example, Dr Bob can only access Alice's data which are stored between 10/02/2019 and 10/06/2019.
- Valid : This is a binary value that determines the validity of the policy, value 1 for valid and value 0 for invalid policies. As the BC ledger is immutable, users cannot change a policy before its expiration time. If a patient requires to revoke permission from a specific user, HW creates the same policy transaction with *"Valid=0"* in BC. The details are explained in Algorithm 1.

Two example policies can be seen in Table 5. In the first policy, the patient with ID A (a 256-bit length) has permitted the medical staff with ID X to access his heartbeat data for two years, from 20/05/2016 to 20/05/2018. This policy is valid until the date 20/11/2020. In the second policy, the same data owner has revoked permission from the medical staff with identifier Y to access all his data from 07/06/2018 to 07/07/2018.

4.2. Data storage and retrieval protocol

This section describes the communication between different components of the BCHealth architecture and user data storage and retrieval processes.

4.2.1. Smartphone to IHM communication

The user's smartphone classifies the received data from sensors based on their type (e.g., heart rate, ECG, EEG, temperature sensor, etc.). A symmetric cryptographic algorithm encrypts this data (e.g., AES) by a key shared between the smartphone and the IHM in the initial registration step. Fig. 7 presents a schematic diagram of communication between the smartphone (or PDA) and the IHM. In the figure, $SK_{PDA,IHM}$ is a shared key between the PDA and the IHM. The ACK message is used as the acknowledgment of receiving data by the IHM.

Fig. 7. PDA and IHM communication.

4.2.2. IHM and BC communication

Upon receiving the health data from sensors, the IHM generates the hash of the data and stores the data (in an internal database). Then sends this hash value as a transaction to the BC network stored in the "Data Chain".

$$Hash = SHA_{256}(Data \| Time \| Type)$$

Where,

- $Data$: received data from the sensors.
- $Time$: date of receiving the data.
- $Type$: type of data.

The communication between the IHM and the BC is depicted in Fig. 8. In this process,

- ID_{own} is the user's ID who generated the transaction.
- ENC_{pr_own} refers to an asymmetric encryption algorithm that uses the data owner's private key.

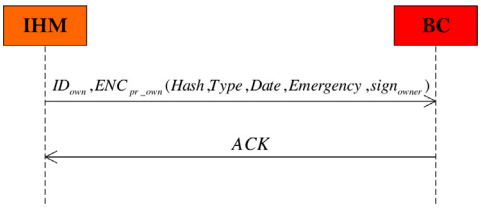

Fig. 8. Communication between the IHM and BC for registering "data transactions".

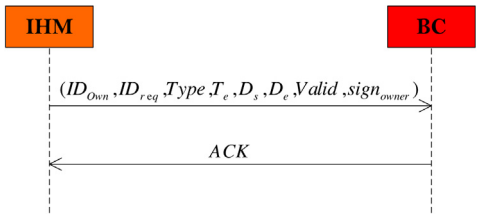

Fig. 9. Communication between the IHM and BC registering "policy transactions".

- $Emergency$ is a binary value that indicates if the input data is emergency or not (one for emergency). Once data is received by IHM, it analyzes them based on the type of data. If the value of the data is higher or lower than the normal range. The emergency value will be turned to one, while otherwise, it will be zero (e.g., a normal heart rate for adults should be between 60 and 100 beats per minute).

Fig. 9 shows the communication between the IHM and BC to submit policy transactions. The notations of the access policy have been explained in Section 4.1.

4.2.3. Access to healthcare data

After storing patient's data in BC, medical staff or any user who needs to access this data should send a request to the BC network. To access patient's data, such as EEG, medical staff sends an access request (i.e., access transaction) to the BC, containing the required information, i.e., data owner's ID, type of data, start and begin date of data. Every miner in the BC which receives this request searches the policy chain to check if the requester (i.e., medical staff) is authorized to access the data. If the requester is authorized, the miner will send a request transaction to the IHM containing the hash of the data, requester ID, and requester address. Finally, the IHM validates the request; if it is a valid request, it sends the user's original data to the medical staff. This whole process is shown in Fig. 10, where:

- ID_{Req} : The data requester's ID.
- ID_{own} : The data owner's ID.
- D_s, D_e : A valid duration of data.
- $Hash_{Data}$: The hash of data.
- Pub_{own} : Data owner's public key.
- Enc_{pr_Req} : Encrypted with requester's private key.
- Enc_{pub_own} : Encrypted with data owner's public key.
- $Sign_{cluster}$: Each cluster has its own signature, which is only known by the nodes inside that cluster so that IHM can verify the messages based on this signature.
- $Address_{Req}$: Address of the data requesting device, e.g., doctor's smartphone address.

4.2.4. Alert to healthcare staff (real-time phase)

We explained in Section 4.2.2, the IHM can specify if a data transaction is an emergency by sending value "1" in the emergency attribute of the data transaction. When a miner receives an emergency transaction,

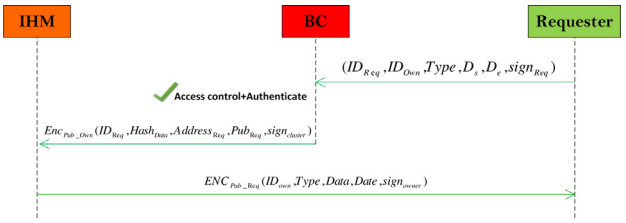

Fig. 10. Data access request process.

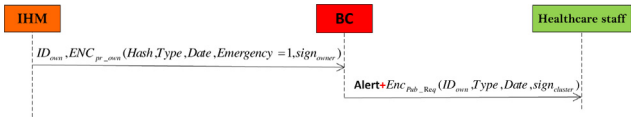

Fig. 11. Communication between the IHM, miner and medical staff for emergency scenario.

it checks the policy chain to find information on the medical staff who are allowed to access this user's emergency data. Then, the miner sends an emergency request to all of these medical staff, notifying them about the patient's emergency situation (refer to Fig. 11).

4.2.5. Access control in miners

One of the main tasks of miners in BCHealth is to control requester's access to user's data based on the defined policy transactions. When a miner receives a request transaction from a user containing the data owner's ID, type of data, start, and the duration of data, it should verify this transaction.

$$(ID_{Req}, ID_{Own}, Type, D_s, D_e, Sign_{Req})$$

If the transaction is valid, the miner searches inside the policy chain for policies that are defined for the requested data. If the requester is authorized, then a hash of the data will be returned to the IHM. This process is presented in Algorithm 1.

Algorithm 1 Access Control Procedure

procedure ACCESS CONTROL
  if ID_{Req} is valid **then** ▷ by checking signature
    $Verify \leftarrow false$
    while all transaction policy with ID_{owner} start search from last block **do**
      if (ID_{Req} in access transaction=ID_{Req} in policy transaction) **AND** ($T_e \leq$ Date Now) **AND** (Type of input equal $'ALL'$ OR Type of input = Type of policy) **AND** (D_s of policy $\leq D_s$ of requester) **AND** (D_e of policy $\geq D_e$ of requester) **then**
        $Verify \leftarrow true$
        Exit while
  if (Verify=true **AND** valid=1) **then** ▷ valid attribute of found transaction was 1
    miner Send $Enc_{pub\_own}(hash_{Data}, Address_{Req}, sign_{Cluster})$ to IHM
  else
    Send "Access Deny" response to the requester

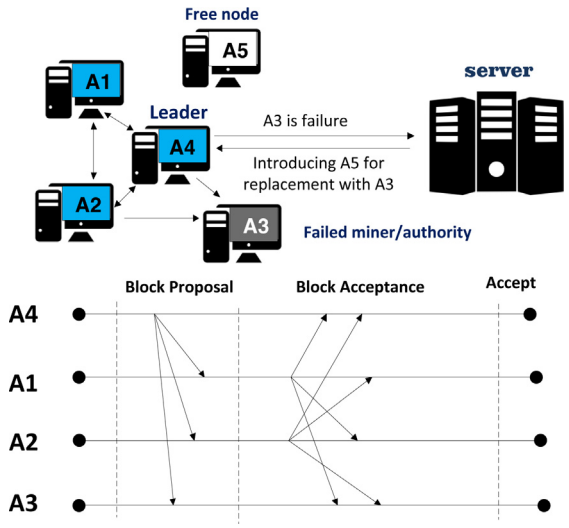

Fig. 12. An example consensus process and node failure recovery.

4.3. Consensus algorithm and miner node availability

This section discusses the consensus algorithm and availability of the miner nodes. As explained in Section 2.1.2, we consider PoA as the consensus algorithm in our BC network. In the PoA algorithm, each cluster has multiple miners or authorities. Time is divided into slots, and in each time slot, one node is considered as the leader. The leader node receives the transactions, generates a block, and sends the block to other authorities (refer to the "block proposal" stage in Fig. 12). In the next phase, the leader and other authorities should confirm or deny the addition of the proposed block to the BC by exchanging messages (refer to the "block acceptance" stage in Fig. 12).

Assume that some nodes are not available anymore (due to a failure or being captured) and cannot participate in the consensus process. We need to address such an issue to guarantee the proper functionality of the BC network (registering and evaluating of blocks). As mentioned in Section 3.5, all miner nodes of a cluster store a lookup table with information about the other nodes of the cluster. In case of a node failure, if the leader does not receive a message from a miner node for several consecutive steps, it can decide to dismiss that corrupted node and notify the corresponding healthcare server. We assume that healthcare servers are aware of all the existing miner nodes in the network, and if requested, they can introduce new miner nodes to the network to be replaced with the corrupted ones. In this case, the server will immediately send the new node's ID to the cluster leader to be introduced as the new node in that cluster. Fig. 12 provides an example of this whole process. In this figure, we consider four miners nodes labeled $A1$ to $A4$ in one cluster. Assume that node $A4$ is selected as the cluster leader. The following steps will be taken by $A4$ to identify and replace a corrupted node.

1. The $A4$ as a leader proposes a block to be added and broadcasts it to all the miners within the cluster.
2. Each node validates the proposed block and sends a confirm or deny message to other nodes.
3. $A4$ (and other nodes) does not receive any message from the node $A3$. Each node in the cluster has a data structure to count the number of consecutive failures of receiving messages from another node. In this example, this value for node $A3$ will be increased to one due to not receiving a message from $A3$, and this will continue up until the number of failed messages from $A3$ reaches a threshold number; e.g., three times in a row. In this case, the leader node consults with other nodes and, after having a consensus, classifies $A3$ as a failed node and notifies the server.

Table 6
Notation Table.

Parameters	Description
PL	Previous leader node
NL	New leader node
$T_{deadline}$	Time threshold to nodes response
T_{max}	Maximum threshold Node non-response

4. The server stores a table including the address of all the valid nodes, as well as the number of reserved nodes per cluster (e.g., node $A5$ is not assigned to any cluster). The server then immediately introduces $A5$ to the cluster leader by sending its ID and address.
5. The leader node validates this identifier and broadcasts it to other nodes within the same cluster.

In order to improve scalability and detect node failure inside each cluster, the leader node executes Algorithm 2 to report the unavailable nodes to the server to replace them with new ones (see Table 6 for notation explanation).

Algorithm 2 Node Failure Procedure

```
1: procedure NODE FAILURE
2:     NL ← array{n, 3} fromPL                                   ▷ column0 = node_id, column1 = counter, column2 = flag
3:     for Each block number do
4:         while current time ≤ T_deadline do
5:             n ← node − id ▷ find node id from message and put array index in n variable
6:             array[n][1] ← 0                                   ▷ Reset failure counter
7:             array[n][2] ← 1                                   ▷ show node participant in block validation
8:         for ar:array do
9:             if ar[2]==0 then
10:                ar[1] ← ar[1] + 1
11:                ar[2] ← 0
12:                if ar[1]> T_max then
13:                    Server ← ar[0]                            ▷ send node_id to server as node failure
```

5. Security analysis

In this section, we discuss the security and privacy aspects of the proposed architecture. We analyze the CIA security triad (Confidentiality, Integrity, and Availability) of our architecture to show its resilience against several attacks. First, we explain the CIA triad aspects in our architecture.

- **Confidentiality:** Confidentiality means that the messages should be accessed only by authorized users. We use symmetric key encryption for communication between the smartphone and the IHM while sending the collected data to preserve confidentiality. Also, the communication between the IHM and BC is secured by using public-key encryption. Moreover, the user data is stored in a secure encrypted database in the IHM, which is only accessible by the IHM.
- **Integrity:** Data integrity ensures that no one can change the stored data without permission. BC inherently is resistant against unauthorized data modification due to the immutability feature of BC. We store the hash of data in IHM and BC. Therefore, in case of an unauthorized modification, these two hash values would not match, leading to manipulation detection.
- **Availability:** Due to the distributed nature of blockchain, BCHealth does not face the usual single point of failure/attack issue; hence, the availability is improved compared to the centralized counterparts. Moreover, as the hash of the data and the

policies are stored in all miner nodes of the same cluster as the user, the availability of the policies and data hash in that cluster is guaranteed (provided that a sufficient number of miner nodes exist in each cluster). In order to ensure the availability of user's healthcare data, which is stored in the IHM, and avoid unavailability of data due to an attack, this data could be replicated on other trusted devices or stored centrally in the corresponding hospital's servers. However, we leave the replication approach for future work while briefly discuss possible options in Section 7.1.

5.1. Attack model

In this section, we explain different attack scenarios that we consider in our attack model and evaluate the resilience of BCHealth against them.

Assumptions and objectives: The main security objective is to protect users' healthcare data against unauthorized access by any third party. As mentioned earlier, the user's healthcare data (D_{ui}) is stored in a user-specific device called IHM. We assume the following trusted, semi-trusted, and malicious components in our system model.

- *Trusted components:* We assume that IHMs are trusted and secure. They are used for storing and analyzing user's data. The D_{ui} will be encrypted before being stored in the IHM_{ui}. Another trusted component in our system is the ministry of health. In BCHealth, policy or data transactions (T_j) between IHM_{ui}, blockchain and authorized medical staff ($\mathcal{P}$) are encrypted by keys that are provided by the ministry of healthcare secret negotiation algorithm.
- *Semi-trusted components:* We assume the healthcare centers, medical staff and miners to be honest but curious. They genuinely follow the protocol in terms of registering users and miners, assigning key materials, defining clusters, and other associated tasks to them. However, they are curious to gain information about patients' data and modify access policies. We use the $H_{\mathscr{A}}$ notation for them.
- Malicious components: We assume an *"outsider attacker"*($\mathscr{A}_{out}$) in our attack model, who is not part of the system and does not have a valid ID, Health Wallet (HW), and key materials. We also assume that there is an *"insider attacker"* ($\mathscr{A}_{in}$), who is a registered user in the system but has been compromised by an adversary. In the rest of the paper, the $\mathscr{A}_{ua}$ notation refers to both the $\mathscr{A}_{out}$ and the $\mathscr{A}_{in}$.

We consider two attack scenarios in our attack model: (1) attack on confidentiality of the health data, and (2) attack on the availability of the system as described in the following subsections.

5.1.1. Unauthorized access to user's data

The $\mathscr{A}_{ua}$ may try one (or more) of the following approaches to gain unauthorized access to the D_{ui}.

- Capturing several miners in a cluster (e.g., M_{C_i}) and trying to modify or add policies.
- Impersonating a medical staff ($\mathcal{P}$) and requesting access to the D_{ui}.
- Eavesdropping the communication between the IHM_{ui} and $\mathcal{P}$ or PDA_{ui} and IHM_{ui} to gain access to the D_{ui}.
- Capturing the IHM_{ui} and trying to decrypt the $Enc(D_{ui})$.

We consider different attack scenarios performed by the honest but curious components, as well as an $\mathscr{A}_{ua}$ through the previously mentioned four approaches and discuss the resilience of BCHealth in each attack scenario.

Scenario 1 (curious healthcare center): We assume a curious healthcare center ($H_{\mathscr{A}}$) trying to gain access to the D_{ui} by controlling some or all of the miners under its supervision. In this scenario, $H_{\mathscr{A}}$ wants to add/modify access policies for the D_{ui}.

Theorem 1. *In the worst-case scenario, the $H_{\mathscr{A}}$ is not able to change policies' chain or disrupt the mining process by controlling all of the miners that are located under its supervising.*

Proof. We considered a set of healthcare centers in our system model as follow:

$$H = \{H_1, \dots, H_n\} \tag{2}$$

each H_i controls several miner nodes, where each of them are allocated to different clusters. If we consider k clusters in the system, and n miner nodes in each cluster, we have:

$$M_{C_i} = \bigcup_{j=1}^{n} M_{jC_i} \quad \text{and} \quad M_{C_{total}} = \bigcup_{i=1}^{k} M_{C_i} \tag{3}$$

where M_{C_i} denotes all the miner nodes in the ith cluster (C_i), M_{jC_i} denotes jth miner node in cluster C_i, and $M_{C_{total}}$ shows all the miner nodes in the whole BC network. Suppose that S_{H_i} is a subset of $M_{C_{total}}$ which is associated with hospital H_i. We consider a condition for assigning miners to different clusters by the Ministry of Health. Maximum one miner node s_k associated with each hospital H_i can be assigned to each cluster M_{C_j} by the Ministry of Health (i.e., each cluster will have only one miner from each healthcare center).

According to the PoA consensus algorithm, $H_{\mathscr{A}}$ needs to control $(N/2+1)$ miners in each cluster to be able to change the transactions in the policy chain. Considering that these miner nodes in each cluster are located in different healthcare centers, we can consider this attack to be a very difficult task for the $H_{\mathscr{A}}$ to compromise $(N/2+1)$ miners in one cluster.

Scenario 2: In this scenario, we assume that the $\mathscr{A}_{in}$ generates a fake transaction T_j to access data D_{ui}. To do this, the attacker must encrypt transaction T_j via user's u_i public key and sends it to IHM_{ui}. So, $\mathscr{A}_{in}$ need to have user's ID, address, and public key to access data D_{ui} as follow:

$$ENC_{pub-ui}(ID_{\mathscr{A}_{in}}, Hash_{Data}, Address_{\mathscr{A}_{in}}, Pub_{\mathscr{A}_{in}}, sign_{cluster})$$

Theorem 2. *BCHealth is resilient against the impersonation/fake transaction attack.*

Proof. For each transaction, the cluster's unique signature needs to be included (as shown in Fig. 10). However, since this signature is not known to the $\mathscr{A}_{in}$, it is identifiable by the IHM that the included signature in this transaction is not genuine and T_j has not been validated by the miners of the medical centers.

Scenario 3: In this scenario, $\mathscr{A}_{ua}$ eavesdrops the transaction exchanged between the IHM_{ui} and the $\mathcal{P}$ or between the user's PDA_{ui} and the IHM_{ui}

Theorem 3. *BCHealth is resilient against the eavesdropping attack.*

Proof. As depicted in Fig. 10, the exchanged messages between the IHM_{ui} and the $\mathcal{P}$ are encrypted by the public key of the $\mathcal{P}$ and can only be decrypted by $\mathcal{P}$'s private key. Therefore, $\mathscr{A}_{ua}$ is unable to decrypt the message and access the D_{ui}.

$$ENC_{pub-\mathcal{P}}(ID_{ui}, Type, Data, Date, sign_{ui})$$

As shown in Fig. 7, the communication between the user's PDA_{ui} and the IHM_{ui} is encrypted using the shared secret key between them, i.e., $SK_{PDA,IHM}(ID_{ui}, Type, Data)$, so confidentiality is preserved.

Scenario 4: In this scenario, an $\mathscr{A}_{out}$ tries to capture an IHM and access the data. The success of this attack depends on the security level of the IHM itself, which will be the patient's responsibility to keep the device secure and consider appropriate security measures.

5.1.2. Denial of service attacks

In this attack, an $\mathscr{A}_{ua}$ intends to disrupt the system by creating a large number of transactions to prevent legitimate users from receiving the required services.

DoS attack by insider attacker: An $\mathscr{A}_{in}$, using her HW, generates and transmits many transactions to the BC. This will lead to the failure of the miner nodes to provide service for authorized users. BCHealth is more robust than a centralized architecture against DoS attacks due to its distributed nature. Since BCHealth uses clustering, transaction flooding will affect only a subset of clusters rather than the whole network. In addition, limiting the number of transactions that each user can send per second would alleviate the effect of the DoS attack.

DoS attack by outsider attacker: An $\mathscr{A}_{out}$ who does not have a valid HW ID can continuously generate fake transactions and use different IDs. BCHealth is resilient against this attack as each miner keeps a list of valid registered IDs belonging to its corresponding cluster. Each miner receives this list from the health centers' servers and can verify the received messages and identify invalid IDs. This helps the miners to validate the requests and identify the invalid IDs.

6. Experimental analysis

Most of the research studies on the usage of blockchain adopt existing open-source tools, such as Hyperledger and Ethereum to implement and evaluate the performance of their proposals [68–70]. However, since we have extensively modified the basic BC structure in BCHealth (i.e., clustered the nodes, considered different topologies, adopted the PoA Consensus Algorithm among the nodes in each cluster, and used two different chains), we could not use the existing tools in our experimental analysis. Therefore, we implemented the BC network from scratch and performed a thorough analysis as reported in this section.

6.1. System setup

We used Python to implement user applications, i.e., HW, for generating transactions. We implemented an application for sending access request transactions and implemented the miners' program on the BC network for receiving and storing transactions, as well as block mining. We used the Mininet network emulator to develop our BC network topology.

We ran Mininet on a PC with a dual-core CPU and 4 GB of RAM. For BC network topology, we considered a network consisting of 12 identical miner nodes that have similar resources and processing power to evaluate and record both policy and data transactions in BC. In addition, we considered four nodes as IHM to generate user's healthcare transactions and send them to the BC network.

6.2. Performance results

To evaluate the performance of BCHealth, we considered the following criteria. The *"transaction registration time"* refers to the amount of time that it takes to record transactions in BC based on the number of clusters while checking policy transactions. We divided 12 miners into one, two, three, and four clusters to measure the effect of the *"size of the cluster"* in transaction registration time. We considered four different scenarios (i.e., no clustering, two, three, or four clusters) to evaluate our architecture's performance. Fig. 13 presents the considered topology for each scenario. We ran each scenario ten times, and for each scenario, we considered 500, 1000, 2000, and 5000 transactions. We then calculated the average time for registering the transactions.

The evaluation results are illustrated in Figs. 14 and 15. As shown in Fig. 14, in the first scenario (no cluster), recording data transactions are much longer than the other three scenarios. This is because all the miners have to store replicated transactions, and broadcasting transactions impose a delay in the network. This time is significantly reduced in the three other clustering cases (2, 3, and 4 clusters). This is because only miners of each cluster store replicated transactions, and different clusters store their transactions in parallel. Therefore, by increasing the number of clusters, more transactions are recorded at the same time. We anticipate that decreasing the number of nodes in each cluster might reduce the registration time due to reducing the broadcasting delay. However, decreasing the number of miners in each cluster increases the risk of malicious activities (less number of nodes require to be compromised by an attacker). Therefore, a significant challenge would be determining the correct number of clusters and the number of nodes in each cluster, which is a trade-off between performance and security. We leave this challenge for future work.

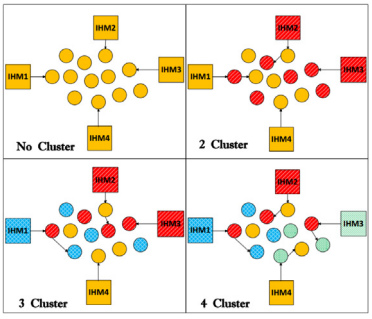

Fig. 13. Topology of BC network that used for simulating BCHealth.

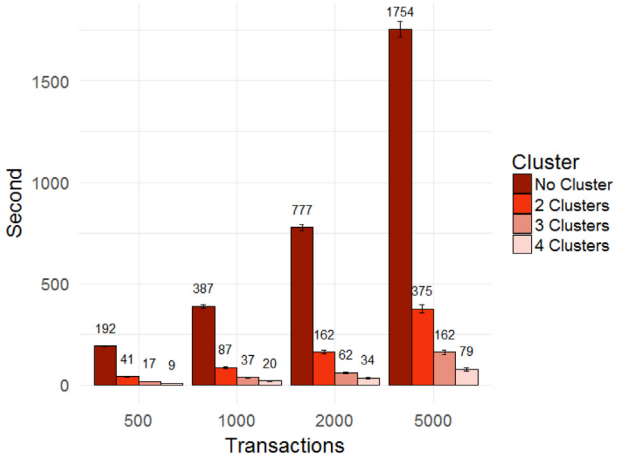

Fig. 14. Data transaction registration time for different number of clusters..

6.3. Performance of access control algorithm

As we mentioned in previous sections, a policy chain is used to record policy transactions in BCHealth. After receiving an access request transaction, each miner needs to verify the request to prevent unauthorized access to the user's data. Therefore, miners need to search for policies in the policy chain to find corresponding policy transactions. In this regard, we evaluated the search time, i.e., the execution time of the access control algorithm. To this end, we sent random transactions to the network and evaluated the time it takes to find the corresponding policy. Fig. 16 shows the results of search time for different numbers of transactions (2000 to 20000 transactions) considering various clustering models. As the number of transactions increases, the search time for finding the target transactions as well.

Table 7
Comparison between BCHealth and related work from security and privacy perspective.

Scheme	Data Security	Data Privacy	Anonymity	Integrity	Authentication	Data leakage prevention
[71]	✓	✓	✓	✓	✓	✘
[72]	✓	✓	✓	✓	✓	✘
[19]	✓	✓	✓	✓	✘	✘
BCHealth	✓	✓	✓	✓	✓	✓

Table 8
Comparison between BCHealth and other blockchain-based EHR storage methods.

Scheme	Data storage method	Description	Advantages	Disadvantages
[73]	Off-chain	Uses healthcare providers' attribute-based signature and stores medical data in a secure third party (Cloud)	1- It is easier for the data owner to locate the healthcare data; 2- No restrictions on block size; 3-secure distributed EHR data sharing	1- Need to trust a third party; 2- The data owner has no control over his/her data
[43]	Off-chain	A medical image sharing framework in which only an authorized requester can access the data.	1- Privacy is better maintained due to access control policies; 2- Access control is base on user satisfaction	1- Single point of failure and the risk of sabotage attacks; 2- Need to trust a third party
[42]	Off-chain	Proposes a blockchain-based platform for exchanging healthcare information.	1- Two fairness-based algorithms for improving the throughput of system 2-Off-chain storage and on-chain verification	1- Sensitive data published on third party servers 2- The data owner has no control over his/her data.
[72]	Cloud	The encrypted data is stored on the Cloud, and the data requester can decrypt the data only if he has enough key shares	Reduced blockchain load	1- Need to trust third parties; 2- Reduced user privacy as the data requester can access the plaintext
[74]	Cloud	Medical data is stored on the Cloud using CP-ABE-based access control (CCAC)	1-Reduced blockchain load; 2- Good access control policy	1- Creating a fully-trusted third party is not easy; 2- It does not resist against attacks to the Cloud
BCHealth	off-chain	User data is stored in IHMs, which are controlled by the data owner. The data owner defines data access policies.	1-No trusted the third party is required; 2-Reduced blockchain load; 3-Data owner decides the data access policies and has control over his/her data	1- Data access time increases slightly (however, a clustering model has been adopted to parallelize the data processing and reduce the access time)

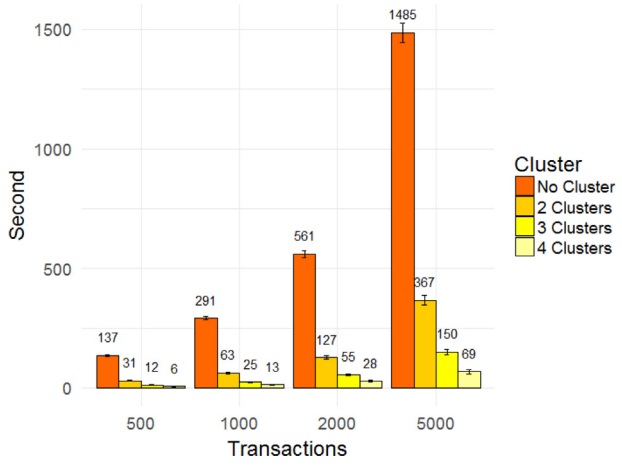

Fig. 15. Policy transaction registration time for different number of clusters.

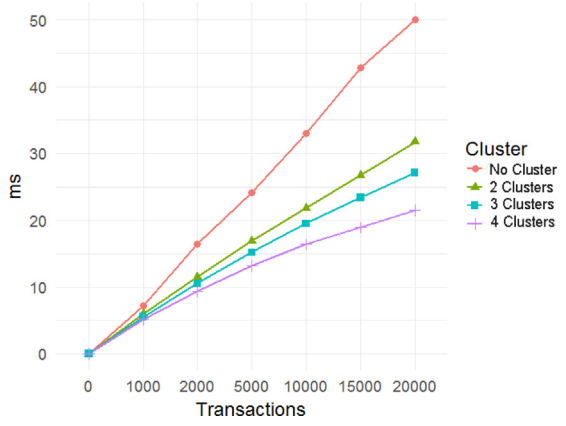

Fig. 16. Search time in policy chain based on number of transactions and clusters..

Increasing the number of clusters will help in reducing this time significantly due to the distribution of policy transactions in different clusters.

6.3.1. Two chains performance

In this section, we show that using two chains (policy and data) in our proposed architecture, reduces the search time of transactions and increases BC network performance. As mentioned before, to access a transaction, first the access policy will be searched in the policy chain, if the access policy exists in the chain, then the data transaction will be checked in the transaction chain. Fig. 17 shows the performance of the proposed approach considering various scenarios. In this figure, the Ch and Cl abbreviations refer to the number of chains and clusters, respectively; while "h" and "m" refer to chain hit or miss, respectively. For example, the 2Ch-3Cl-h scenario refers to a BC network with 2 chains and 3 clusters where the policy chain is hit, so the data chain is searched to find the data transaction. In this figure, the number of policy transactions is a quarter of the data transactions (e.g., for 10,000 data transactions, we generate 2,500 policy transactions). As it can be seen in this figure, in the case of using a single chain, on average the $n/2$ chain transactions should be looked up which increases the transaction access time significantly. However, with the introduction of two chains and reducing the policy chain size, searching the data chain when the policy requirement is not met can be avoided. Consequently, this will improve the performance of the BC network in terms of transaction access time.

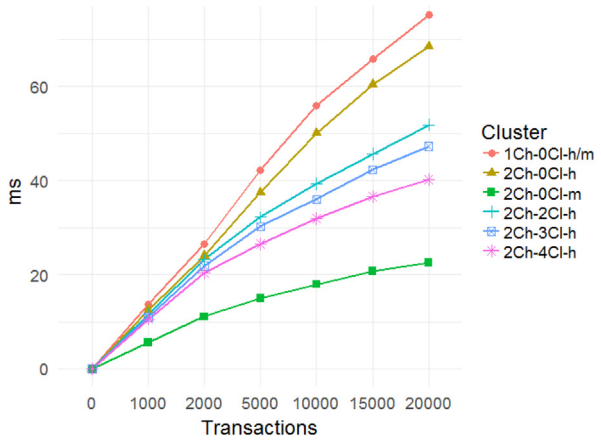

Fig. 17. Search time considering one chain and two chains (i.e., policy and transaction chains).

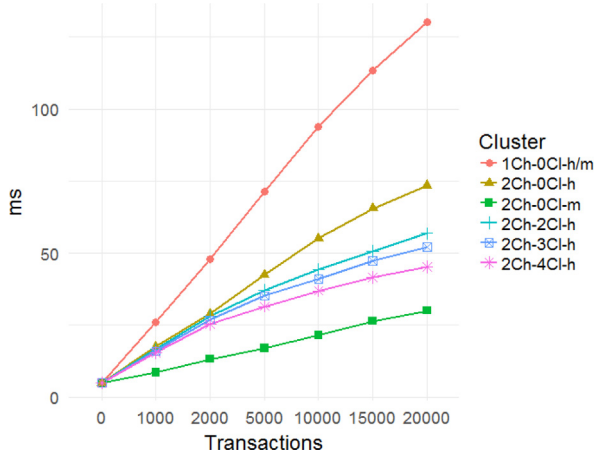

Fig. 18. Total required time to access user data from when the request is issued to when the data has been delivered. Various scenarios are considered using one chain or two chains (i.e., policy and transaction chains) and several clustering models.

6.3.2. Access time

We define the access time as the time that takes for the medical staff to access the data from when the request has been sent to the BC until when the data has been received, this includes:

- The time that is required for the request to be sent to the BC network
- The search time in the policy and data chains (shown in Fig. 17).
- The time that is required for the data to be sent from the BC to the IHM (the required time for validating and finding data in the IHM is neglected due to the storage of this data in the main memory of IHM).
- The time that is required for the data to be sent from the IHM to the medical staff.

Fig. 18 shows access time having considered several scenarios with different cluster numbers and chains. We ignored the time that is required for mining BC blocks. This figure compares single chain BC with the proposed two chains model for different numbers of clusters to demonstrates how access time has been improved remarkably. When the policy chain is missed, this means that the data is not accessible and therefore there is no need to search the data chain, so the access time is significantly reduced (as can be seen in scenario 2Ch-0CL-m). Also, in scenarios with two clusters, it is obvious that increasing the number of clusters will decrease the access time.

7. Discussion

This section discusses data availability in IHM and then compares BCHealth with the state-of-the-art from two different perspectives, data security and data storage, to show its outperformance.

7.1. Data availability in IHM

Regarding data availability in the IHM, we could consider two options for backup in case of emergency: (1) the user's trusted people's IHM (including family or close friends), whose address is specified by the users in the initialization stage, and (2) centrally store it in the healthcare center. The latter contradicts our whole idea of decentralization. Therefore, if we consider the former option, one may argue that this backup data could be prone to attacks or misuse. One possible solution for that could be adopting Secret Sharing methods [75]. We leave this discussion for future work.

7.2. Data security

As discussed earlier, health data has been considered sensitive information, storing it in the blockchain; a public ledger might trigger data security and privacy concerns. Several solutions have been proposed in blockchain to address this concern, which could be classified as cryptographic or non-cryptographic mechanisms, including anonymization and access control methods. A blockchain-based platform has been proposed in [71] that uses Elliptic Curve Cryptography to encrypt personal data. In [72] symmetric key encryption has been used to encrypt and upload the data on the Cloud. A blockchain-based application known as Healthcare Data Gateway (HDG) has been proposed in [19], which calculates and stores the encrypted data directly on the private blockchain cloud network without exposing the raw data. These methods increase data security and privacy, while they do not provide access control mechanisms and the data owner does not have any control over his/her data [76]. Table 7 presents a comparison between BCHealth and the related work. In this table data leakage prevention column refers to ensuring the confidentiality of the sensitive user data. In BCHealth, only the hash of the data is stored on the blockchain, and the raw data is stored on each user's IHM, where the user has full access control over it. Therefore the probability of data leakage is very low compared to the related work where they store this sensitive data on the blockchain network or Cloud.

7.3. Health data storage

There are two main methods for storing EHRs in blockchain, storing the medical data (raw or hash) directly in the blockchain or the off-chain method. The first approach imposes high computational and storage overhead. Moreover, since duplicate data will be stored in all the blockchain network nodes, the probability of data leakage increases. To solve these issues, some researchers [72,74,77] introduced the *off-chain* method. In this approach, the original data is encrypted and stored by a trusted third party, such as Cloud, while the metadata is stored in the blockchain for verification and data integrity purposes. Table 8 compares BCHealth and some related work discussing their strengths and weaknesses.

8. Conclusion

In this paper, we proposed BCHealth architecture for preserving user's privacy in healthcare applications. We propose a blockchain-based approach that enables users to share their healthcare data with medical staff while having control over accessing their data. To increase scalability and throughput, we grouped the BC network nodes into several clusters and assigned each user to a specific cluster for storing his data and access policies. We also proposed a hierarchical approach

based on user-assigned IDs to speed up search and access to specific clusters. We discussed several attack scenarios and the resilience of BCHealth against these attacks. Finally, we implemented our own BC network using Python and Mininet and evaluated the performance of the proposed architecture. Our analysis results report significant improvement in terms of network delay and cost.

A challenging issue in our work (and all the clustering-based literature) is cluster management and optimizing the number of clusters and the number of nodes in each cluster. We leave the cluster management optimization for future work. Another challenge, which we left for future work, is defining a load balancing method for healthcare centers to assign the best cluster to each user based on the network load and location of the user. Moreover, we will look into integrating Cloud in our architecture for storing archive data.

CRediT authorship contribution statement

Koosha Mohammad Hossein: Conceptualization, Methodology, Writing – original draft. **Mohammad Esmaeil Esmaeili:** Methodology, Data curation, Experimental analysis, Writing – original draft. **Tooska Dargahi:** Supervision, Writing – review & editing. **Ahmad Khonsari:** Supervision, Reviewing. **Mauro Conti:** Reviewing and editing.

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.

References

[1] T.M. Fernández-Caramés, P. Fraga-Lamas, A review on the use of blockchain for the internet of things, IEEE Access 6 (2018) 32979–33001.

[2] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the Internet of Things: A comprehensive survey, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1676–1717.

[3] J. Best, IoT and the NHS: Why the internet of things will create a healthcare revolution, 2018, https://www.zdnet.com/article/iot-and-the-nhs-why-the-internet-of-things-will-create-a-healthcare-revolution/.

[4] E. Petelko, IoT in healthcare: Use cases, trends, advantages and disadvantages, 2019, https://medium.com/existek/iot-in-healthcare-use-cases-trends-advantages-and-disadvantages-8213e738e03.

[5] E. Gultepe, J.P. Green, H. Nguyen, J. Adams, T. Albertson, I. Tagkopoulos, From vital signs to clinical outcomes for patients with sepsis: a machine learning basis for a clinical decision support system, J. Am. Med. Inform. Assoc. 21 (2) (2013) 315–325.

[6] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi, Internet of Things security: A survey, J. Netw. Comput. Appl. 88 (2017) 10–28.

[7] H. Wang, Z. Li, Y. Li, B.B. Gupta, C. Choi, Visual saliency guided complex image retrieval, Pattern Recognit. Lett. 130 (2020) 64–72.

[8] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, Z. He, Self-supervised deep correlation tracking, IEEE Trans. Image Process. 30 (2020) 976–985.

[9] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, C. Zhang, Compound rank-k projections for bilinear analysis, IEEE Trans. Neural Netw. Learn. Syst. 27 (7) (2015) 1502–1513.

[10] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in: Internet of Everything, Springer, 2018, pp. 103–130.

[11] H. Li, H. Huang, S. Tan, N. Zhang, X. Fu, X. Tao, A new revocable reputation evaluation system based on blockchain, Int. J. High Perform. Comput. Network. 14 (3) (2019) 385–396.

[12] C. Esposito, M. Ficco, B.B. Gupta, Blockchain-based authentication and authorization for smart city applications, Inf. Process. Manage. 58 (2) (2021) 102468.

[13] Y.-A. De Montjoye, E. Shmueli, S.S. Wang, A.S. Pentland, Openpds: Protecting the privacy of metadata through safeanswers, PLoS One 9 (7) (2014) e98790.

[14] K.N. Griggs, O. Ossipova, C.P. Kohlios, A.N. Baccarini, E.A. Howson, T. Hayajneh, Healthcare blockchain system using smart contracts for secure automated remote patient monitoring, J. Med. Syst. 42 (7) (2018) 130.

[15] A. Ekblaw, A. Azaria, J.D. Halamka, A. Lippman, A Case Study for Blockchain in Healthcare:"MedRec" prototype for electronic health records and medical research data, in: Proceedings of IEEE Open & Big Data Conference, Vol. 13, 2016, p. 13.

[16] X. Liang, J. Zhao, S. Shetty, J. Liu, D. Li, 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, 2017, pp. 1–5.

[17] M. Mettler, Blockchain technology in healthcare: The revolution starts here, in: 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services (Healthcom), IEEE, 2016, pp. 1–3.

[18] J. Zhang, N. Xue, X. Huang, A secure system for pervasive social network-based healthcare, Ieee Access 4 (2016) 9239–9250.

[19] X. Yue, H. Wang, D. Jin, M. Li, W. Jiang, Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control, J. Med. Syst. 40 (10) (2016) 218.

[20] T.-T. Kuo, L. Ohno-Machado, Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks, 2018, arXiv preprint arXiv:1802.01746.

[21] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 839–858.

[22] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, G. Das, Everything you wanted to know about the blockchain: Its promise, components, processes, and problems, IEEE Cons. Electron. Magaz. 7 (4) (2018) 6–14.

[23] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Lsb: A lightweight scalable blockchain for iot security and privacy, 2017, arXiv preprint arXiv:1712.02969.

[24] A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, Medrec: Using blockchain for medical data access and permission management, in: 2016 2nd International Conference on Open and Big Data (OBD), IEEE, 2016, pp. 25–30.

[25] C. Sullivan, E. Burger, E-residency and blockchain, Comput. Law Secur. Rev. 33 (4) (2017) 470–481.

[26] A.D. Dwivedi, G. Srivastava, S. Dhar, R. Singh, A decentralized privacy-preserving healthcare blockchain for iot, Sensors 19 (2) (2019) 326.

[27] A. Dorri, S.S. Kanhere, R. Jurdak, Towards an optimized blockchain for IoT, in: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, ACM, 2017, pp. 173–178.

[28] A.D. Dwivedi, L. Malina, P. Dzurenda, G. Srivastava, Optimized blockchain model for internet of things based healthcare applications, 2019, arXiv preprint arXiv:1906.06517.

[29] J. Konečnỳ, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[30] K. Mohammad Hossein, M.E. Esmaeili, T. Dargahi, A. Khonsari, Blockchain-based privacy-preserving healthcare architecture, in: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), IEEE, 2019, pp. 1–4.

[31] J. Al-Jaroodi, N. Mohamed, Blockchain in industries: A survey, IEEE Access 7 (2019) 36500–36515.

[32] C. Roulin, A. Dorri, R. Jurdak, S. Kanhere, On the activity privacy of blockchain for IoT, 2018, arXiv preprint arXiv:1812.08970.

[33] M. Vukolić, The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication, in: International Workshop on Open Problems in Network Security, Springer, 2015, pp. 112–125.

[34] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Proj. Yellow Pap. 151 (2014) (2014) 1–32.

[35] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, Pbft vs proof-of-authority: Applying the CAP theorem to permissioned blockchain, Univ. Southampt. Inst. Repos. (2018).

[36] A. team of engineers, Parity technologies, 2019, https://www.parity.io.

[37] A. team of engineers, Go ethereum., 2013-2019, https://geth.ethereum.org.

[38] Sphinx, Proof-of-authority consensus, 2018, https://apla.readthedocs.io/en/latest/concepts/consensus.html.

[39] Y. Zhang, Y. Sun, R. Jin, K. Lin, W. Liu, High-performance isolation computing technology for smart IoT healthcare in cloud environments, IEEE Internet Things J. (2021).

[40] J.L. Shah, H.F. Bhat, A.I. Khan, Integration of cloud and IoT for smart e-healthcare, in: Healthcare Paradigms in the Internet of Things Ecosystem, Elsevier, 2021, pp. 101–136.

[41] A. Yazdinejad, G. Srivastava, R.M. Parizi, A. Dehghantanha, K.-K.R. Choo, M. Aledhari, Decentralized authentication of distributed patients in hospital networks using blockchain, IEEE J. Biomed. Health Inf. 24 (8) (2020) 2146–2156, http://dx.doi.org/10.1109/JBHI.2020.2969648.

[42] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, J. He, Blochie: a blockchain-based platform for healthcare information exchange, in: 2018 Ieee International Conference on Smart Computing (Smartcomp), IEEE, 2018, pp. 49–56.

[43] V. Patel, A framework for secure and decentralized sharing of medical imaging data via blockchain consensus, Health Inform. J. 25 (4) (2019) 1398–1411.

[44] R. Gupta, A. Shukla, P. Mehta, P. Bhattacharya, S. Tanwar, S. Tyagi, N. Kumar, Vahak: A blockchain-based outdoor delivery scheme using uav for healthcare 4.0 services, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2020, pp. 255–260.

[45] R. Kumar, N. Marchang, R. Tripathi, Distributed off-chain storage of patient diagnostic reports in healthcare system using IPFS and blockchain, in: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2020, pp. 1–5.

[46] P. Zhang, J. White, D.C. Schmidt, G. Lenz, S.T. Rosenbloom, Fhirchain: applying blockchain to securely and scalably share clinical data, Comput. Struct. Biotechnol. J. 16 (2018) 267–278.
[47] G.G. Dagher, J. Mohler, M. Milojkovic, P.B. Marella, Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, Sustainable Cities Soc. 39 (2018) 283–297.
[48] B.B. Gupta, K.-C. Li, V.C. Leung, K.E. Psannis, S. Yamaguchi, et al., Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system, IEEE/CAA J. Autom. Sin. (2021).
[49] A.P. Mohan, A. Gladston, et al., Merkle tree and blockchain-based cloud data auditing, Int. J. Cloud Appl. Comput. 10 (3) (2020) 54–66.
[50] G.N. Nguyen, N.H. Le Viet, M. Elhoseny, K. Shankar, B. Gupta, A.A. Abd El-Latif, Secure blockchain enabled Cyber–physical systems in healthcare using deep belief network with ResNet model, J. Parallel Distrib. Comput. 153 (2021) 150–160.
[51] C.L. Stergiou, K.E. Psannis, B.B. Gupta, IoT-based big data secure management in the fog over a 6G wireless network, IEEE Internet Things J. (2020).
[52] A. Dorri, S.S. Kanhere, R. Jurdak, Blockchain in internet of things: challenges and solutions, 2016, arXiv preprint arXiv:1608.05187.
[53] P.J. Taylor, T. Dargahi, A. Dehghantanha, R.M. Parizi, K.-K.R. Choo, A systematic literature review of blockchain cyber security, Digit. Commun. Netw. (2019).
[54] G. Zyskind, O. Nathan, et al., Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, IEEE, 2015, pp. 180–184.
[55] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, S. Liu, Blockchain-based data preservation system for medical data, J. Med. Syst. 42 (8) (2018) 141.
[56] K. Fan, S. Wang, Y. Ren, H. Li, Y. Yang, Medblock: Efficient and secure medical data sharing via blockchain, J. Med. Syst. 42 (8) (2018) 136.
[57] T. Dey, S. Jaiswal, S. Sunderkrishnan, N. Katre, HealthSense: A medical use case of internet of things and blockchain, in: 2017 International Conference on Intelligent Sustainable Systems (ICISS), IEEE, 2017, pp. 486–491.
[58] L. Ismail, H. Materwala, S. Zeadally, Lightweight blockchain for healthcare, IEEE Access 7 (2019) 149935–149951.
[59] A.D. Dwivedi, G. Srivastava, S. Dhar, R. Singh, A decentralized privacy-preserving healthcare blockchain for IoT, Sensors 19 (2) (2019) 326.
[60] G. Srivastava, R.M. Parizi, A. Dehghantanha, The future of blockchain technology in healthcare internet of things security, in: Blockchain Cybersecurity, Trust and Privacy, Springer, 2020, pp. 161–184.
[61] T. Verbelen, P. Simoens, F. De Turck, B. Dhoedt, Cloudlets: Bringing the cloud to the mobile user, in: Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services, ACM, 2012, pp. 29–36.
[62] Z. Sheng, C. Mahapatra, C. Zhu, V.C. Leung, Recent advances in industrial wireless sensor networks toward efficient management in IoT, IEEE Access 3 (2015) 622–637.
[63] E. Fombu, The Future of Healthcare: Humans and Machines Partnering for Better Outcomes, 2018.
[64] S. Sen, L. Datta, S. Mitra, Machine Learning and IoT: A Biological Perspective, CRC Press, 2018.
[65] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in: 2017 IEEE International Congress on Big Data (BigData Congress), IEEE, 2017, pp. 557–564.
[66] H.-N. Dai, Z. Zheng, Y. Zhang, Blockchain for internet of things: A survey, 2019, arXiv preprint arXiv:1906.00245.
[67] M. Zamani, M. Movahedi, M. Raykova, Rapidchain: Scaling blockchain via full sharding, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 931–948.
[68] A. Khatoon, A blockchain-based smart contract system for healthcare management, Electronics 9 (1) (2020) 94.
[69] H.L. Pham, T.H. Tran, Y. Nakashima, A secure remote healthcare system for hospital using blockchain smart contract, in: 2018 IEEE Globecom Workshops (GC Wkshps), IEEE, 2018, pp. 1–6.
[70] M. Li, L. Xia, O. Seneviratne, Leveraging standards based ontological concepts in distributed ledgers: a healthcare smart contract example, in: 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), IEEE, 2019, pp. 152–157.
[71] A. Al Omar, M.S. Rahman, A. Basu, S. Kiyomoto, Medibchain: A blockchain based privacy preserving platform for healthcare data, in: International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Springer, 2017, pp. 534–543.
[72] X. Zheng, R.R. Mukkamala, R. Vatrapu, J. Ordieres-Mere, Blockchain-based personal health data sharing system using cloud storage, in: 2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom), IEEE, 2018, pp. 1–6.
[73] Y. Sun, R. Zhang, X. Wang, K. Gao, L. Liu, A decentralizing attribute-based signature for healthcare blockchain, in: 2018 27th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2018, pp. 1–9.
[74] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, M. Guizani, BPDS: A blockchain based privacy-preserving data sharing for electronic medical records, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–6.
[75] A. Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.
[76] S. Shi, D. He, L. Li, N. Kumar, M.K. Khan, K.-K.R. Choo, Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey, Comput. Secur. (2020) 101966.
[77] A. Juneja, M. Marefat, Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification, in: 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), IEEE, 2018, pp. 393–397.