This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

Healthchain: A Blockchain-based Privacy Preserving Scheme for Large-scale Health Data

Jie Xu, Kaiping Xue, *Senior Member, IEEE,* Shaohua Li, Hangyu Tian,
Jianan Hong, Peilin Hong, Nenghai Yu

***Abstract*—With the dramatically increasing deployment of the Internet of Things (IoT), remote monitoring of health data to achieve intelligent healthcare has received great attention recently. However, due to the limited computing power and storage capacity of IoT devices, users' health data are generally stored in a centralized third party, such as the hospital database or cloud, and make users lose control of their health data, which can easily result in privacy leakage and single-point bottleneck. In this paper, we propose Healthchain, a large-scale health data privacy preserving scheme based on blockchain technology, where health data are encrypted to conduct fine-grained access control. Specifically, users can effectively revoke or add authorized doctors by leveraging user transactions for key management. Furthermore, by introducing Healthchain, both IoT data and doctor diagnosis cannot be deleted or tampered with so as to avoid medical disputes. Security analysis and experimental results show that the proposed Healthchain is applicable for smart healthcare system.**

***Index Terms*—Blockchain, privacy preserving, dynamic key management, Internet of Things (IoT), smart healthcare, fine-grained access control.**

I. Introduction

THE Internet of Things (IoT) is an emerging and promising technology that connects a large number of smart devices to the Internet, where devices collect and exchange data to help people monitor changes and respond to them to improve efficiency [1, 2]. Currently, it has been applied in many fields, such as vehicle network [3], smart grid industry [4], smart home [5], in which, by leveraging IoT technology, smart healthcare has received more and more attentions.

IoT technology based smart healthcare has been proposed to significantly improve efficiency and accuracy, break geographical restrictions to achieve remote monitoring [6], conduct disease risk assessment [7], and construct disease prediction systems [8]. In smart healthcare system, IoT devices, such as wearable sensors, keep collecting users' physiological data, such as electrocardiogram (ECG), blood pressure, temperature and so on. Usually, these physiological data are sent to the user's local gateway to perform further data processing, aggregation, and then sent to a healthcare provider for diagnosis and feedback, so that users can further better understand their own health status. However, these personal smart health devices are characterized by miniaturization and ultra-low power consumption, resulting in limited computing and storage capacity [1]. Therefore, smart health devices require additional methods to assist in computing and storage. So far, a common approach is to outsource personal health data and electronic health records (EHRs) to cloud servers [7].

Cloud-assisted healthcare system improves efficiency and reduces cost compared with traditional health system. However, it should be noted that there are still many drawbacks in the system: 1) Large-scale smart health devices require high computing and storage capabilities of cloud servers. Since cloud storage and computing can also be seen as centralized to a certain extent, once cloud servers break down or are attacked, all users might be affected. 2) Health data is highly sensitive and should be well protected. Cloud server may leak user privacy for commercial benefits. For example, users only allow their health data to be accessed by authorized professional healthcare staffs, but cloud providers may leak users' personalized EHRs, for medical research, drug advertising and so on, without the user's permission for increasing their own benefits [9]. 3) When a medical dispute occurs, the user may suspect that the original EHRs stored in the cloud has been modified as the distrust of the third party. Besides, it is difficult to share data stored in cloud among different platforms with specific access control policies.

The blockchain technology provides a public, digitized and distributed ledger, which is firstly proposed by Nakamoto [10]. It has been widely used in cryptocurrency transactions such as Bitcoin [10] and Ether [11]. Meanwhile, it has also become the key technology for various IoT scenario for more innovations. All nodes in the blockchain construct a Peer-to-Peer (P2P) network to interconnect with each other. All participating nodes are equal and collaboratively provide services without a single central point, which can avoid the risk of single-point bottleneck. The blockchain consists of a series of blocks and grows over time, in which each block mainly contains a hash of its previous block, a timestamp, a nonce, and some transactions. A transaction records the data that a user wants to add to the blockchain, and new transactions are broadcast to other nodes. Some nodes collect new transactions into a block. The method to add a block to a blockchain is determined by a specific consensus mechanism. Nodes accept the block only if all transactions in it are valid. Once a block is added to the blockchain, it cannot be tampered with under certain security assumptions. The blockchain cannot be forked and all nodes keep working on its extension.

In this paper, we propose Healthchain, a blockchain-based privacy preserving scheme for health data. In Healthchain, users can periodically upload the health data collected by IoT

J. Xu, K. Xue, S. Li, H. Tian, P. Hong and N. Yu are with Department of EEIS, University of Science and Technology of China, Hefei, Anhui 230027, China (Email: kpxue@ustc.edu.cn, K. Xue).

J. Hong is with Huawei Shanghai Research Institute, Shanghai 201206, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

devices and publish them as a transaction. Doctors or artificial intelligence (AI) health analyzers can diagnose anytime and anywhere based on the IoT data and publishes the diagnosis as a transaction. In fact, with the explosive growth of the Internet of Things devices, there will be large-scale health data and these health data will continue to increase. It is not appropriate to record users' complete data on the blockchain, as resource requirements for each node on the blockchain will be extremely high. Otherwise, the blockchain will be too complex to maintain, search and verify. Considering the limited storage capacity of each blockchain node, we introduce InterPlanetary File System (IPFS), which is a content-addressable, distributed file system to store data with high integrity and resiliency. There is no central server in IPFS, and data are distributed and stored in different IPFS nodes all over Internet. Thus, IPFS has no single point of failure. IPFS can efficiently distribute large amounts of data without duplication [12]. Each file uploaded to the IPFS system has a unique hash string through which the file can be retrieved. In our proposed Healthchain, users' complete health data is stored in IPFS storage system. Only hash string of health data, stored in blockchain, is used to verify data's integrity and map to the complete data in IPFS storage. In this way, Healthchain supports large-scale health data and has good scalability.

However, in addition to storage pressure of the massive data, the issue of data security and user privacy is also one of the major challenges. On one hand, the open and transparent nature of the blockchain makes users' privacy easy to be compromised. On the other hand, authorized professional healthcare providers, e.g., doctors or AI health analyzers, need to access users' health data. Therefore, users' health data should be encrypted and fine-grained access control should be conducted over the encrypted data. Only authorized professional healthcare providers can get specific users' health data. In order to enhance the security protection of health data, Healthchain allows users to update encryption keys, revoke and add authorized professional healthcare providers at any time.

The main contributions of this paper can be summarized as follows:

1) We propose a blockchain-based smart healthcare system for large-scale health data privacy preserving, named Healthchain. In Healthchain, users are enabled to upload IoT data and read doctors' diagnoses, and meanwhile, doctors are allowed to read users' IoT data and upload diagnose. In addition, all IoT data and diagnoses cannot be tampered with or denied, which can avoid medical disputes.
2) Healthchain separates transactions for publishing data from transactions for fine-grained access control, and meanwhile data is encrypted and stored in IPFS (Inter-Planetary File System), which can efficiently reduce communication overhead and computation overhead while ensuring privacy preserving.
3) Furthermore, by uploading updated transactions about security keys, Healthchain can allow users to dynamically revoke doctors and update keys at any time.

The rest of this paper is organized as follows. In Section II, we review the related work. The system model, threat model and design goals are introduced in Section III. In Section IV, we describe the details of our proposed scheme. The security analysis and performance evaluation are given in Section V and Section VI. Finally, Section VII concludes this paper.

II. Related Work

In this section, we discuss the related works in terms of traditional smart healthcare system, blockchain application in network scenarios and smart healthcare based on blockchain.

A. Traditional smart healthcare system

Nowadays, people are increasingly hoping to get more accurate, comprehensive and efficient health information about themselves, and meanwhile their personal privacy can be well preserved. With the development of information and communication technology (ICT), and cloud computing, many research efforts have been devoted to improving the efficiency and security of smart healthcare systems.

To protect personal health data stored in semi-trusted cloud servers, attribute-based encryption (ABE) is introduced to achieve fine-grained access control [13]. In 2013, Li *et al.* [14] proposed a novel patient-centric framework for fine-grained and scalable data access control by using ABE technology to encrypt users' EHR data. In order to solve the problem of revealing access policies in traditional Ciphertext-policy attribute-based encryption (CP-ABE), Zhang *et al.* [13] proposed to hide the specific and sensitive attribute values in the access policy. Recently, Zhang *et al.* [15] analyzed and found that there are a large amount of duplicate EHR data in the cloud storage. In order to reduce the storage cost in cloud servers, in [15], they further proposed an effective solution to allow cloud servers to remove duplicate data and reduce storage costs. Hua *et al.* [16] proposed CINEMA, which is an effective, privacy-preserving primary diagnostic framework for online healthcare, in which, based on the fast secure permutation and comparison technologies, users can implement query operations on cloud servers without decrypting their private data. However, CINEMA requires cloud servers to have high computing and storage performance to enable millions of users to query online at the same time.

Although these schemes provide secure storage and fine-grained access control in cloud, there are still some problems existing in the systems, such as how to prevent internal malicious attacks and cloud server crashes. Therefore, in this paper, we introduce a distributed blockchain-based system instead of cloud servers for data storage and privacy protection.

B. Blockchain application in network scenarios

Blockchain has been originally proposed for constructing a public distributed ledger for all transactions in Bitcoin [10]. After that, many research efforts focus on key problems of the blockchain technology itself, such as performance improvement [17, 18], solving the double spending attack [19, 20] and constructing efficient distributed consensus mechanisms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

[21, 22]. Meanwhile, there are also many other researches which focus on developing blockchain-based practical applications. In addition to acting as the infrastructures for cryptocurrency systems [11], it can also be integrated to many IoT scenarios.

For example, in the vehicular networks, to effectively evaluate the trustworthiness of vehicles in non-trusted environments, Yang *et al.* [22] proposed a decentralized trust management system based on blockchain techniques to update and publish the trust information of all the vehicles in vehicular networks. They also improved distributed consensus by proposing a new consensus mechanism to compete for updated trust for all RSUs. Compared to [22], Kang *et al.* [23] utilized smart contracts to store and share vehicular data for efficient automated data management. In smart grid, in order to realize optimal scheduling and protect user's private information, Guan *et al.* [24] proposed a blockchain-based privacy-preserving and efficient data aggregation scheme, where users are divided into different groups and for each group a user is selected as a miner to aggregate the data in the group and adds it to the group's private blockchain. However, these schemes can address their stated issues in the specific network scenarios, but they cannot be straightly adopted in smart healthcare systems. In smart healthcare, for privacy preserving, not only user's IoT data, but also doctor's diagnosis should be protected. In particular, from the perspective of the participants, although the user can be anyone, in order to ensure the safety of the users, the doctors who diagnose users need to be examined for eligibility. Therefore, we propose Healthchain, which includes a Userchain and a Docchain to achieve privacy protection in smart healthcare.

C. Smart healthcare based on blockchain

In recent years, many studies have shown that blockchain is a promising solution to achieve personal health information security and privacy protection. Some research efforts [25–27] devote to demonstrating the advantages of smart healthcare systems based on blockchain and propose architectures, but lack specific implementation details. Some literatures, such as [28, 29], focus on fine-grained access control of IoT data collected from users. However, they do not further consider the privacy protection of electronic medical records (EMRs) generated by the doctors. In addition, some schemes [30–34] are dedicated to utilizing blockchain technology to enable users to control their EMRs, which are controlled by the hospital in traditional smart healthcare systems. Al *et al.* [30] presented a user centric healthcare data privacy preserving scheme called MediBchain. In MediBchain, users encrypt sensitive health data and store them on permissioned blockchain. Only users with the correct password can get data from MediBchain. However, users must share passwords when sharing their health data, which can conduct a coarse-grained access control, but it may lead to key leaks easily. MediBchain lacks password update and key update schemes. Moreover, MediBchain is vulnerable to replay attacks and offline dictionary attacks. After that, Zhang *et al.* [31] utilized Shamir's secret sharing to authenticate users and doctors for fine-grained access authorization. However, in Zhang *et al.*'s scheme, EMRs are stored in a blockchain, and the blockchain is maintained in a trusted cloud, which leads to centralization. The same problem exists in Yue *et al.*'s [32] scheme. The literatures [30–32] can achieve health data mastered by users, but as the number of users and the volume of health data increase, due to the limited size of blocks, these schemes may lead to intolerable authentication delay and storage. In order to reduce the user's storage overhead and improve the throughput of the blockchain, in [33], medical records are stored in external databases, and the pointers to external databases for medical records and reading permissions are stored in smart contract on the Ethereum blockchain. Recently, Dagher *et al.* [34] proposed to use blocks to store hash values of medical records while sending the actual query link information in a private transaction over HTTPS. However, this method is vulnerable to DoS attacks.

In addition to the problems pointed out above, there are still difficulties in key management and flexible revocation. Therefore, we propose Healthchain, which not only supports fine-grained access control for large-scale data, but also implements key management and flexible revocation using independent key transactions.

III. System Model, Threat Model and Design Goals

In this section, we introduce the system model, threat model and design goals of a blockchain-based smart healthcare architecture, named Healthchain.

A. System model

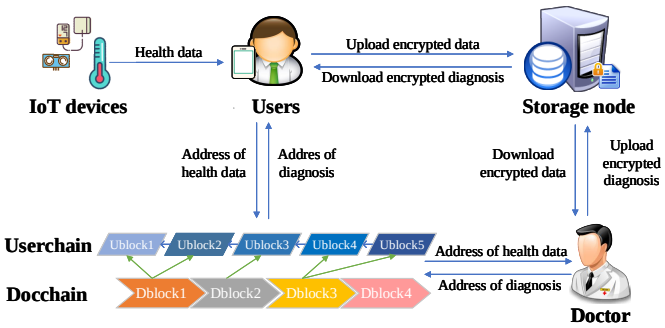


Fig. 1. System model of Healthchain

As shown in Fig. 1, Healthchain can be divided into several different components, which are described in details as follows:

- ***IoT devices***. They may be wearable sensors or implanted sensors. IoT devices monitor users' health parameters such as weight, heart rate, calories burned, sleep patterns, blood glucose levels, and so on. Each IoT device has one and only one user node as its management node. They send various collected health-related data to the user node periodically. IoT devices are characterized by portability, low power, and personalization, and limited

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

computing and storage capabilities. Thus, they are not directly involved in the blockchain.

- ***User nodes***. Each one U_i is the management of one or more IoT devices, which can aggregate, encrypt data from IoT devices and send them to the storage node. There are many lightweight user nodes that only store the block headers of Userchain, and they can only generate and publish transactions. Meanwhile, there are also some user nodes with strong computing and storage capabilities, called core user nodes. They can store complete Userchain, which is defined below. Core user nodes can generate, publish, verify and assist lightweight user nodes to search transactions. They can also mine new Ublocks and add new user transactions to a new Ublock. In addition, all user nodes can also implement information search on Docchain but cannot add transactions on Docchain.
- ***Doctor nodes***. Each one D_j can be not only a real doctor from a hospital, but also an artificial intelligence health analyzer from a smart healthcare service company. They can provide continuous diagnosis based on users' IoT data. All hospitals and companies in Healthchain form a consortium, and all doctor nodes' behaviors is restricted by the rules of the consortium. Authorized doctor nodes can read the information on Userchain and generate transactions for Docchain. Specially, doctor nodes themselves cannot add transactions to Docchain.
- ***Accounting node***. It's a special node in the system, which is deployed by the consortium. It can verify that whether the transactions from doctor nodes are correct and valid. At each time period, all accounting nodes select a leader. The leader aggregates valid transactions from doctor nodes in the consortium, and generates new Dblock and adds new Dblock to Docchain.
- ***Storage nodes***. They collaboratively store complete encrypted users' IoT data and encrypted doctors' diagnoses in a distributed manner. In this paper, we assume that each storage node is IPFS-based, where IPFS system is managed and maintained by the consortium of healthcare providers, e.g., hospitals. IPFS uses a content addressing method where the address is derived from the content of the file. Each file is hashed into a hash string and each hash string is unique to identify the file. Anyone can find the complete file stored in IPFS via the hash string of the file on Userchain or Docchain. IPFS makes it possible to distribute high volumes of data with high efficiency.
- ***Userchain***. It's a public blockchain, which is used to publish users' data. Anyone can join Userchain to read transactions, send transactions, and mine at any time. Userchain consists of a series of Ublocks and grows over time. Each Ublock contains the hash of the previous Ublock and transactions generated by users.
- ***Docchain***. It's a consortium blockchain, which is used to publish doctors' diagnoses. Only doctor nodes authorized by consortium can generate diagnosis transactions, which can be added to Docchain by the accounting nodes. However, anyone can read the information on Docchain. Docchain consists of a series of Dblocks and grows over time. Each Dblock contains the hash of the previous Dblock and transactions doctors generated.

As illuminated in Fig. 1, here we briefly show data flows in our scheme: IoT devices send health data to the user node periodically or on event triggers. The user node encrypts the IoT data and sends them to an IPFS storage node. User node adds the hash of the encrypted data as a transaction to Userchain. The doctor node decrypts the users' data and gives real-time online diagnoses. Then the doctor sends the encrypted diagnosis to the storage node and generates a transaction for diagnosis which includes the address of the encrypted diagnosis. Users read the information on Docchain to understand their own health status.

B. Threat model

We assume that there is a secure channel between the IoT device and the user node. The doctor nodes strictly enforce the specification and give the diagnoses honestly. The private keys of users and doctors are secure in storage. We introduce distributed IPFS nodes for storage, and by using encryption, users' and doctors' data can be securely and stably stored. There are active adversaries and passive adversaries in the system, where passive adversaries eavesdrop on communication channels to get all transmitted data and active adversaries attempt to tamper with or delete messages from users or doctors.

In addition, we assume that all adversaries cannot control more than 51% of the core user node that can generate new Ublocks in Userchain. We assume that there are $3f + 1$ accounting nodes in the consortium, of which there are no more than f malicious nodes.

C. Design goals

We aim to achieve privacy-preserving for intelligent medical systems, and the following design goals should be met.

- ***Supporting large-scale IoT devices***: It is estimated that there will be more than 24 billion connected IoT devices all over the world by 2020 [35]. For smart healthcare, more and more IoT devices continue to generate health data, which brings challenges to system design. Therefore, the system needs to be able to process massive data generated by massive IoT devices and further support devices' dynamically joining and exiting.
- ***High efficiency***: The large amounts of health data needs to be stored and analyzed timely and securely. Real-time online diagnosis is also very important, which can even save the lives of users. Therefore, the user's health data is uploaded in time and read with specific access policies. Similarly, the doctor's diagnosis also needs to be uploaded in time and accurately and read by the user.
- ***Privacy-preserving***: Each user's health data can be only obtained by himself/herself and his/her authorized professional healthcare staff (doctors, AI health analyzers, etc.). Meanwhile, doctor's diagnosis can be accessed by the diagnosed user and the authorized professional healthcare staffs. No adversary can get the user's private information.
- ***Accountability***: In order to prevent medical disputes, the doctor needs to be responsible for the diagnosis he/she has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

made and cannot tamper with or deny it. Anyone can audit whether past diagnoses have been tampered with.
- ***On-demand revocation***: The user can revoke the right of a doctor to access his/her IoT data at any time. The revoked doctor cannot read the data after revocation, which is called forward security.

IV. Proposed Scheme: Healthchain

In this section, we first give the overview of our proposed efficient privacy preserving for smart healthcare system.

A. Overview

To achieve both non-tampering of IoT data and diagnosis, as shown in Fig. 1, Healthchain consists of two sub-blockchains, respectively named as Userchain and Docchain.

Userchain is introduced to ensure that users' transactions cannot be tampered with by anyone including the users themselves. There are two types of user transactions on Userchain: IoT transactions and key transactions. IoT transactions are used to protect the integrity of IoT data, and key transactions are used for access control. The main part of an IoT transaction is a hash of encrypted IoT data, which can be used to address encrypted IoT data at IPFS nodes. The main part of a key transaction is two symmetric keys: one called IoT key for encrypting/decrypting IoT data and the other called diagnosis key for encrypting/decrypting diagnosis. Both symmetric keys are generated by the user and encrypted with the authorized doctor's public key. The authorized doctor node can obtain two symmetric keys to decrypt users' IoT data or encrypt diagnosis by decrypting the key transaction. IoT transactions and key transactions are generated independently, and users can generate them based on their needs. Core user nodes add users' transactions to Userchain.

There is only one type of transactions in Docchain called diagnosis transaction, which are encrypted with users' diagnosis key. In order to generate a diagnosis transaction, the authorized doctor node first searches Userchain for transactions of the users they are responsible for. If the transaction found is a key transaction, the doctor node updates the stored keys for encrypting/decrypting IoT data or diagnosis. If it is related to IoT data, the doctor node goes to the IPFS system to get the complete IoT data based on the hash of user's IoT data in the IoT transaction. Then, the doctor node generates corresponding diagnosis for the user based on the IoT transactions in a timely manner. The doctor node encrypts the diagnosis and stores it to IPFS system. The doctor node further generate a transaction including a hash of the encrypted diagnosis, and then broadcasts the diagnosis transaction to nodes involved in Docchain. Accounting nodes collect diagnosis transactions and add them to Docchain. By leveraging blockchain technology, Docchain can ensure that diagnosis transactions cannot be tampered with by anyone.

Therefore, our scheme implements privacy preserving of users' health data and conducts fine-grained access control with Userchain and Docchain.

B. Details of our proposed scheme

In the following, we give a detailed introduction of our proposed system, which can be divided into five layers. As shown in Fig. 2, from bottom to top, these five layers are given as: *Data layer*, *Network layer*, *Consensus layer*, *Incentive layer*, and *Application layer*.

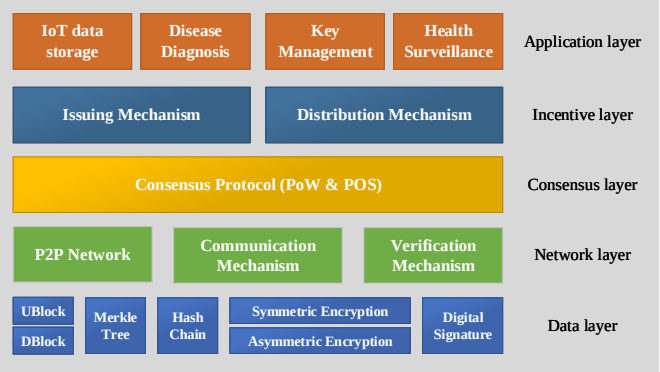


Fig. 2. The Architecture of Healthchain

1) ***Data layer****:* The data layer is at the bottom. There are two main data structures in the data layer: Ublock and Dblock, and a few cryptographic algorithms.
- **Ublock**. Userchain consists of Ublocks, where each Ublock contains information about users. As seen from Fig. 3, each Ublock can be divided into two main parts: block header and block body. The block header contains an index $Index$, a timestamp $Gtime$, a hash of the previous block $prehash$, a nonce $nonce$, and a root of the merkle tree $userroot$. The merkle tree, as the block body in a Ublock, contains hash values of user transactions.

To protect the privacy of users, we use a symmetric encryption algorithm, such as AES, to encrypt IoT data in users' transactions. In the existing schemes, encrypted data and a corresponding secret key protected in a digital envelope are usually combined together and sent to the authorized receivers. Different from this way, in order to reduce the overhead for doctors to decrypt digital envelopes, we decouple the encrypted data and the corresponding keys, respectively in the form of IoT transactions and key transactions. Users can update key transactions as needed instead of updating the key each time the IoT transaction is updated. The user keeps using the key contained in the current key transaction. Therefore, users can update keys more flexibly. It is worth mentioning that the key transaction also contains the key for encrypting the doctor' diagnosis generated directly by the user. We record the symmetric key for diagnosis encryption as diagnosis key, and the symmetric key for IoT data encryption as IoT key.

Therefore, there are two types of transactions for Ublock: transactions about IoT data tx_{IoT}, and transactions about keys tx_{key}. Users generate tx_{IoT} to transmit encrypted IoT data to authorized doctors and generate tx_{key} to flexibly adjust the authorization for doctors, such as adding or revoking authorized doctors. In addition, if the IoT key or diagnosis key is compromised, the user can update it at any time by generating and transmitting a new key transaction tx_{key}. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

latest tx_{key} contains updated IoT key and diagnosis key that are encrypted separately with all currently authorized doctors' public keys.

$$\begin{aligned} &tx_{IoT} = \{ID_{U_i}, ts_1, HEIoT, S_i, htxI_i\}, \\ &where \\ &S_i = Sign(sk_{U_i}, H(ID_{U_i}, ts_1, HEIoT)), \\ &htxI_i = H(ID_{U_i}, ts_1, HEIoT, S_i). \end{aligned} \tag{1}$$

As in Eq. (1), a transaction of IoT data tx_{IoT} contains the identity of the user ID_{U_i}, who publish the transaction, timestamp of the transaction ts_1, hash of the encrypted IoT data $HEIoT$, the signature S_i signed with the specific user's private key sk_{U_i}, and $htxI_i$, which is the hash of all the other parts in the transaction. Besides, $htxI_i$ is the identity of the transaction, and is a leaf node of the merkle tree, which makes it more efficient for users to find a specific transaction. It is noteworthy to include $htxI_i$ in tx_{diag} to denote the corresponding IoT data as the cause of diagnosis. In fact, $htxI_i$ is a link between Userchain and Docchain, and each diagnosis transaction is associated with multiple IoT transactions, further reducing the possibility of medical disputes. The symmetric key used to encrypt IoT data is Ik_i. Specially, in order to reduce user's storage overhead, only the hash $HEIoT$ of the encrypted IoT data is in the transaction instead of the completed encrypted IoT data. Users can obtain a corresponding hash string $HEIoT$ by uploading encrypted IoT data $Enc(Ik_i, IoT)$ to the IPFS system. Anyone can get completed encrypted IoT data from IPFS storage nodes based on $HEIoT$.

$$\begin{aligned} &tx_{key} = \{ID_{U_i}, ts_2, Env_{ij}, Env_{U_i}, Sig_i, htxk_i\}, \\ &where \\ &Env_{ij} = \{ID_{D_j}, htxI_i, Enc\{pk_{D_j}, (Ik_i, dk_{ij})\}\}, \\ &Env_{U_i} = Enc\{pk_{U_i}, (Ik_i, dk_{ij})\}, \\ &Sig_i = Sign(sk_{U_i}, H(ID_{U_i}, ts_2, Env_{ij}, Env_{U_i})), \\ &htxk_i = H(ID_{U_i}, ts_2, Env_{ij}, Env_{U_i}, Sig_i). \end{aligned} \tag{2}$$

As in Eq. (2), a transaction about session key tx_{key} contains the identity of the user ID_{U_i}, who publish the transaction, the identity of current authorized doctor node ID_{D_j}, identity of IoT transaction $htxI_i$ that keys contained in the key transaction can decrypt, timestamp of the transaction ts_2, the encrypted updated key, the signature Sig_i signed with specific user's private key sk_{U_i}, and $htxk_i$, which is the hash of other all the other parts in the transaction. It should be noted that $htxk_i$ is the identity of the transaction and also is the first layer of the merkle tree, which makes it more efficiently for users to find a specific transaction. Specially, the encrypted updated key contains two types of digital envelopes, one for authorized doctors and the other for user. Digital envelope for each authorized doctor ID_{D_j} contains the current IoT key Ik_i and diagnosis key dk_{ij} encrypted with the doctor's public key pk_{D_j}. Digital envelope for the user contains the current IoT key Ik_i and diagnosis key encrypted with user's public key pk_{U_i}. Therefore, both authorized doctors and the user can obtain the IoT key or diagnosis key by searching the key transaction. It should be pointed out that an IoT key Ik_i may decrypt several encrypted IoT data, and a user can enjoy health service from several doctor nodes at the same time. When a user updates an IoT encryption key Ik_i, in order to increase the efficiency of key update, several doctor identities and digital envelopes can be included in a key transaction. When a user needs to revoke a doctor, he/she only needs to generate a new key transaction, which contains updated digital envelopes containing a new IoT key to authorized doctors.

Because the genesis block of Userchain is the first block of Userchain, it does not contain the previous block hash. The genesis Ublock contains the identity of the genesis Ublock $Index$, a timestamp $Gtime$, a nonce $nonce$, the root of the merkle tree $userroot$, and genesis users' transactions.

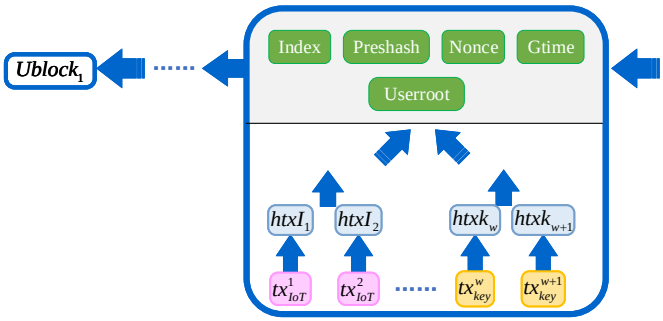


Fig. 3. The structure of Userchain

- **Dblock**. Docchain is composed of Dblocks. Similarly to Ublock, as shown in Fig. 4, each Dblock can be divided into two main parts: block header and block body. The block header contains an index $Index$, a timestamp $Gtime$, a hash of the previous block $prehash$, a nonce $nonce$, and the root of the merkle tree $diagroot$. The merkle tree, as the block body in a Dblock, contains hash values of diagnosis transactions.

$$\begin{aligned} &tx_{diag} = \{ID_{D_j}, ts_3, htxI_i, HEdm, S_j, htxd_j\}, \\ &where \\ &S_j = Sign(sk_{D_j}, H(ID_{D_j}, ts_3, htxI_i, HEdm)), \\ &htxd_j = H(ID_{D_j}, ts_3, htxI_i, HEdm, S_j). \end{aligned} \tag{3}$$

As in Eq. (3), a transaction of diagnosis tx_{diag} contains the identity of the doctor ID_{D_j}, who publish the transaction, timestamp of the transaction ts_3, the identifier of the user's IoT transactions $htxI_i$, hash of encrypted diagnosis $HEdm$, signature S_j signed by doctor D_j, and $htxd_j$, which is the hash of all other parts in the transaction. In addition, $htxd_i$ is the identity of the transaction and is also the first layer of the merkle tree, which makes it more efficiently for users to find a specific transaction. It is important to highlight that the IoT data associating with $htxI_i$ is the cause of the corresponding diagnosis and $htxI_i$ is also a link between Userchain and Docchain. If the doctor generates a diagnosis based on several tx_{IoT}, the diagnosis transaction contains several corresponding $htxI_i$. In this way, the diagnosis produced by the doctor bases on the corresponding IoT data, which can further reduce the possibility of medical disputes. The symmetric key used to encrypt diagnosis $diag$ is dk_{ij}, which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

obtained by decrypting the digital envelope in key transaction. Specially, in order to reduce the doctor's storage overhead, only the IPFS hash of the encrypted diagnosis $HEdm$ is in the transaction instead of the completed encrypted diagnosis. Doctor can obtain the corresponding hash string $HEdm$ by uploading encrypted diagnosis $Enc\{dk_{ij}, diag\}$ to the IPFS system. Anyone can get the completed encrypted diagnosis $Enc\{dk_{ij}, diag\}$ from the IPFS based on $HEdm$.

Because the genesis block of Docchain is the first block of Docchain, it does not contain previous block hash. The genesis Dblock contains the identity of the genesis Dblock $Index$, a timestamp $Gtime$, a nonce $nonce$, a root of the merkle tree $diagroot$, and doctors' genesis transactions.

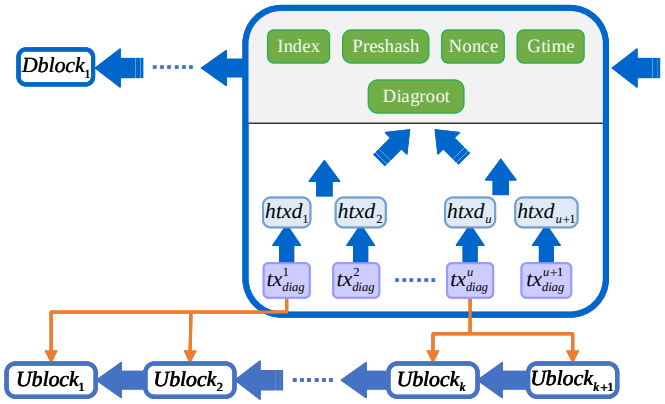


Fig. 4. The structure of Docchain

*2) **Network layer**:* The second layer is the network layer. Blockchain is a peer-to-peer network based on the Internet. In Healthchain, there are IoT devices, user nodes, doctor nodes, storage nodes, accounting nodes and others in the network. Each IoT device has one and only one user node as its management node. IoT device periodically sends the data it collects to its management node. After receiving IoT data, user node aggregates and encrypts the data. The complete encrypted data is sent to an IPFS storage node and the hash of the encrypted data, which is the address of the encrypted data, is added to the user's transaction. User node broadcasts the transaction to other user nodes it knows in the network. If core user nodes in the network receive the transaction, they firstly verify whether the signature in the transaction is correct, whether the structure of the transaction is correct, whether the size is within the specified range and so on. If all the verification is successful, the transaction will be further aggregated and added to a new Ublock.

There are several accounting nodes in Docchain that are deployed by the consortium. They act as miners to aggregate transactions generated by doctor nodes. At each time period, all accounting nodes select a leader to add the new Dblock to Docchain. The result of selected leader is broadcast to all accounting nodes and doctor nodes. When a diagnosis is generated, diagnosis is encrypted and sent to IPFS storage node. Then, the doctor generates a diagnosis transaction. The diagnosis transaction is firstly sent to the leader of accounting nodes for the current time period. The leader of accounting nodes first verifies whether the signature in the transaction is correct, whether the transaction is generated by a legitimate doctor node, whether the structure of the transaction is correct, whether the size is within the specified range and so on. If all the verification is successful, the transaction will be broadcast to other accounting nodes. After the accounting node verifies the transaction, it broadcasts its verification result to all accounting nodes including the leader. The leader collects the results of the transaction verification from the other accounting nodes. If all the verifications are successful, the leader aggregates the transactions and records them in the new Dblock. Specifically, the IPFS is led by the consortium.

*3) **Consensus layer**:* Since blockchain is a peer-to-peer network, each node may receive different transactions at a certain time. The consensus mechanism determines when and which node adds a new block to the blockchain for the transaction it receives. Because Userchain is a public blockchain, and Docchain is a consortium blockchain, the two blockchains have their own consensus in Healthchain.

- **Userchain**. Since Userchain is a public blockchain, anyone can send and aggregate transactions. A malicious user node may masquerade as several user nodes at a low cost, known as Sybil attack. However, other users cannot distinguish whether it is a Sybil node or a real user. This makes it difficult to fairly select a core user node to add a new block to Userchain. We choose the consensus mechanism of Proof of Work (PoW) to select a core user node to aggregate users' transactions, generate a new Ublocks and add it to Userchain. Through PoW, a core user node can prove that it has certain capabilities, and it is a legitimate user node rather than a Sybil node. Core user node continues to generate nonce until a nonce is found to satisfy $H(nonce||prehash||userroot) < target$, before the other core user node successfully generates a new Ublock. It is noteworthy that $target$ is dynamically changeable to adjust the speed of new block generation. Our scheme sets the generation period of the block to 1 minute. Thus, $New\ target = Old\ target * (Actural\ time\ of\ last\ 2016\ blocks/2016\ minutes)$. **Algorithm 1** shows the detail of the PoW in Healthchain. Any user who successfully adds a new Ublock to Userchain can have a Healthcoin. Healthcoin is the token of our system, representing a certain amount of work in Healthchain. Its specific use is described in the incentive layer IV-B4.
- **Docchain**. Since Docchain is a consortium blockchain, only accounting nodes authorized by the consortium can aggregate transactions generated by permissioned doctors and add Dblock to Docchain. Instead of relying on the computationally intensive consensus mechanism PoW, as shown in Fig. 5, we choose Practical Byzantine Fault Tolerance (PBFT) [36] as the consensus of Docchain.

We assume that there are a total of $3f+1$ accounting nodes in the consortium. There is only one leader in each time period, which is rotated by accounting nodes. Each accounting node broadcasts the transactions sent from doctor nodes to the whole network. After the leader receives transactions, the leader first sorts the transactions and assigns serial numbers to the transactions. Then, the leader stores the transactions and serial

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

Algorithm 1: Consensus algorithm for Userchain

Input: Hash of the previous block $prehash$, collected users' transactions tx_{IoT} or tx_{key} $Tx = [tx_1, tx_2, ..., tx_n]$, current difficulty value $target$;
Output: Nonce value $nonce$;

Set $userroot = BlockMerkleRoot(Tx)$;
Initialise $nonce = 0$;
Initialise $H_{temp} = \infty$;
Initialise $par = 0$;
while $H_{temp} \geq target$ *and* $par = 0$ **do**
  $nonce++$;
  $H_{temp} = H(nonce||prehash||userroot)$;
  if *Received new Ublock others generated* **then**
    $par = 1$;
  end
  if $H_{temp} < target$ *and* $par = 0$ **then**
    return $nonce$;
  else
    Continue;
  end
end

numbers in its log, and multicasts a PRE-PREPARE message with the transactions and sequence numbers to other accounting nodes. After receiving transactions from the leader, each accounting node verifies whether the signatures, timestamps, sequence numbers etc. are valid. If valid, the accounting node multicasts the PREPARE message containing the signature of authentication result. If an accounting node receives more than $2f$ PREPARE messages from different nodes within a specific time range, it indicates that the PREPARE phase has been completed and the accounting node multicasts a COMMIT message to other accounting nodes. If an accounting node receives more than $2f+1$ different commit messages including itself, it considers that the COMMIT phase is complete and all accounting nodes have reached a consensus to record these transactions to a new Dblock. Finally, the accounting node returns the corresponding reply to the doctor node who generated the transaction. If the consensus fails, change the leader, and restarts the PRE-PREPARE phase once again.

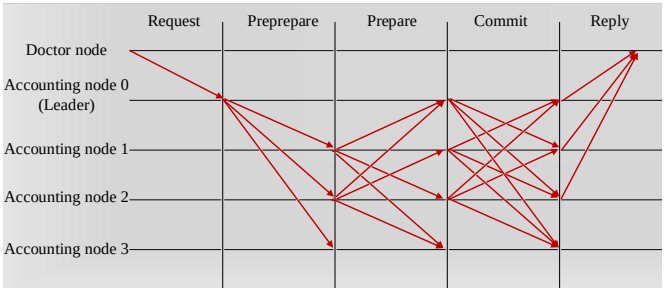


Fig. 5. The overview of message flows in PBFT protocol instances

*4) **Incentive layer**:* In order to promote more users to continue to participate in Healthchain, economic factors are considered in the incentive layer. Considering that Userchain is a public blockchain, as many schemes [18, 23] do, we introduce Healthcoin to Userchain as an incentive token. On the one hand, any node with sufficient capability can act as a core user node for mining, executing the **Algorithm 1** to find the correct $nonce$ to get Healthcoin. Miners can exchange the Healthcoin into any currency they want, such as bitcoin, ether, etc., at the trading center. On the other hand, users need to exchange their own currency for Healthcoin at the trading center to access smart healthcare services. On the premise that the user gives enough Healthcoin to the consortium, the consortium provides smart healthcare services to the user. The user generates IoT transaction to Userchain, and generates the key transaction to authorize the doctors in the consortium. Healthcoin is consumed when the doctor's diagnosis transaction is successfully added to Docchain. Doctor node gets rewards from the consortium based on transactions he/she adds to Docchain.

*5) **Application layer**:* The topmost application layer provides different services for users and doctors. Specifically, IoT data security, key management, and disease diagnosis can be provided in our scheme.

IoT data security. After receiving the latest IoT data from IoT devices, the user node periodically encrypts the IoT data and generate IoT transactions. **Algorithm 2** shows the generation of tx_{IoT} for user U_i. The generated IoT transactions are broadcast to other user nodes. Finally, the transaction are added to Userchain by a core user node.

Algorithm 2: IoT data security

Input: U_i's IoT key Ik_i, IoT data IoT;
Output: Transaction tx_{IoT};

foreach *IoT data upload time slot* **do**
  Encrypt IoT data $EIoT_i = Enc(Ik_i, IoT)$;
  Send $EIoT_i$ to the IPFS storage nodes and get $HEIoT_i$;
  Generate timestamp ts_1;
  Set $S_i = Sign(sk_{U_i}, H(ID_{U_i}, ts_1, HEIoT_i))$;
  Set $htxI_i = H(ID_{U_i}, ts_1, HEIoT_i, S_i)$;
  Set $tx_{IoT} = \{ID_{U_i}, ts_1, HEIoT_i, S_i, htxI_i\}$;
  return tx_{IoT};
end

Key Management. Since doctor nodes may be compromised and leak users' key, the user needs to be able to revoke a doctor at any time and in time. In addition, depending on the need of the user, the user may need to add doctors dynamically. By publishing a new key transaction, our scheme allows users to dynamically add or revoke doctors at any time. When the user establishes contact with a new doctor, the user generates a key transaction contains the current IoT key and a diagnosis key issued to the additional doctor. When a user needs to revoke a doctor, the user first generates a new IoT key. Then, the user publishes a new key transaction, which only contains digital envelopes issued to the currently authorized doctors. Digital envelopes contain the updated IoT key. Therefore, the revoked doctor does not have the new IoT key, and can no longer read the user's data. Besides, user regularly updates the IoT key to prevent offline dictionary attacks. Key

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

transactions and IoT transactions decoupling can reduce both communication overhead and computational overhead for both users and doctors. The steps in **Algorithm 3** are implemented in order to achieve both key management and dynamic doctor enrollment or revocation.

Algorithm 3: Key management

Input: User U_i's public key pk_{U_i}, the public key $pk_{D_1}, ..., pk_{D_j}$ of all current authorized doctors $D_1, ..., D_j$, the identity of U_i's IoT transaction $htxI_i$ that the key contained in the digital envelopes can decrypt;

Output: Transactions tx_{key};

foreach *key update* **do**
  Generate new IoT key Ik_i;
  foreach *authorized doctor* D_j **do**
    Generate new diagnosis key dk_{ij};
    Set $Env_{ij} = \{ID_{D_j}, htxI_i, Enc\{pk_{D_j}, (Ik_i, dk_{ij})\}\}$;
  end
  Generate timestamp ts_2;
  Set $Env_{U_i} = Enc\{pk_{U_i}, (Ik_i, dk_{i1}, .., dk_{ij})\}$;
  Set U_i's signature of the updated keys $Sig_i = Sign\{sk_{U_i}, H(ID_{U_i}, ts_2, Env_{i1}, ..., Env_{ij}, Env_{U_i})\}$;
  Set $htxk_i = H(ID_{U_i}, ts_2, Env_{i1}, ..., Env_{ij}, Env_{U_i}, Sig_i)$;
  Set $tx_{key} = \{ID_{U_i}, ts_2, Env_{i1}, ..., Env_{ij}, Env_{U_i}, Sig_i, htxk_i\}$;
  return tx_{key};
end

Disease diagnosis. The doctor node continuously detects whether there is a transaction with ID_{U_j} on Userchain, which is the identity of the user they are responsible for. Once detected, the doctor first goes to the consortium to check whether the user has paid enough Healthcoin. If so, the doctor performs the following steps. If the transaction is a tx_{key}, then the doctor updates the user key in time. If the transaction is tx_{IoT}, the doctor uses the hash contained in the IoT transaction to IPFS storage node to obtain complete IoT data. Then, the doctor node gives the corresponding diagnosis based on the IoT data. Next, the doctor generates a diagnosis transaction. **Algorithm 4** illustrates the process of the doctor generating a diagnosis transaction. Finally, diagnosis transaction is sent to the accounting nodes and added to Docchain.

Algorithm 4: Disease diagnosis

Input: Identity of the cause for diagnosis $htxI_i$, doctor D_j, diagnosis $diag$ and diagnosis key dk_{ij};

Output: Transaction tx_{diag};

Encrypted diagnosis $Edm_j = Enc(dk_{ij}, diag)$;
Send Edm_j to the IPFS storage nodes and get $HEdm_j$;
Generate timestamp ts_3;
Set the signature of the diagnosis $S_j = Sign(sk_{D_j}, H(ID_{D_j}, ts_3, htxI_i, HEdm_j))$;
Set $htxd_j = H(ID_{D_j}, ts_3, htxI_i, HEdm_j, S_j)$;
Set $tx_{diag} = \{ID_{D_j}, ts_3, htxI_i, HEdm_j, S_j, htxd_j\}$;
return tx_{diag};

V. Security Analysis

In this section, we analyze the security of Healthchain based on the design goals defined in section III-C.

A. Privacy preserving

The user's IoT data and the doctor's diagnosis are very sensitive, and need to be inaccessible to illegal adversaries. As described in the data layer section IV-B1, Userchain only contains the hash of encrypted IoT data $HEIoT$ and adversaries can only get $Enc(Ik_i, IoT)$ from IPFS. IoT data is encrypted with the IoT key Ik_i. Ik_i is encrypted with the doctor's public key pk_{D_j} or the user's public key pk_{U_i}. We assume that the adversaries' computing power is limited, and user's private key sk_{U_i} and doctor's private key sk_{D_j} are secure. Adversaries cannot get Ik_i without sk_{U_i} or sk_{D_j}. Without the key Ik_i, adversaries cannot get the IoT data. Therefore, our scheme could provide conditional security of IoT data.

Similarly, Docchain only contain the hash of encrypted diagnosis $HEdm$ and adversaries can only get $Enc(dk_{ij}, diag)$ from IPFS. The diagnoses is encrypted with the diagnosis key dk_{ij}. The diagnosis key dk_{ij} is encrypted with doctor's public key pk_{D_j} or the user's public key pk_{U_i}. We assume that adversaries' computing power is limited, and user's private key sk_{U_i} and doctor's private key sk_{D_j} are secure. Adversaries cannot get dk_{ij} without sk_{U_i} or sk_{D_j}. It is worth noting that even the doctor $D_{j'}$ who is authenticated by the user U_i can't get the dk_{ij}. Without the key dk_{ij}, no one can get the diagnoses. Thus, our scheme could provide conditional security of diagnoses.

B. Accountability

Accountability means that any third party can audit whether the IoT data is generated by a user and a diagnosis is made by a doctor. On the one hand, users should be responsible for their IoT data. Because the user's transactions tx_{IoT} and tx_{key} both contain the user's signature, under the assumption that the user's private key sk_{U_i} is secure, no one can impersonate a user to generate transactions without sk_{U_i}. Once malicious data is detected, the corresponding user can be found according to the signature contained in the transaction. Therefore, malicious data generated by a user to consume medical resources of the entire system is undeniable.

On the other hand, in order to avoid medical disputes, doctors should be responsible for the diagnoses. We assume that authorized doctors make accurate diagnoses, and their private keys are secure. Because diagnosis transaction tx_{diag} contains the cause of the diagnosis $htxI_i$ and the timestamp of the diagnosis, which are hashed and signed by the doctor, no one can impersonate a doctor to generate the transaction. Since all diagnoses are recorded on Docchain, they cannot be modified according to the assumptions of the threat model defined in section III-B. If the doctor fails to make the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

appropriate diagnosis in accordance with professional rules, he/she needs to be held accountable. Therefore, the proposed scheme is accountable.

C. Revocability

If a user is dissatisfied with a doctor, the doctor can be revoked. In order to revoke a doctor, the user generates a new tx_{key}, which only contains digital envelopes for the other authorized doctors. More precisely, the user first generates a new IoT data key Ik_i'. Then, the user encrypts the new Ik_i' with the other authorized doctors' public keys. Miners add the new tx_{key} to the Userchain. The subsequent IoT data is encrypted with the new Ik_i', so the revoked doctor cannot obtain the user's IoT data any more. Therefore, our scheme successfully provides revocability.

VI. Performance Evaluation

In this section, we experiment to validate the effectiveness and feasibility of Healthchain. This section can be further divided into three parts. In the first part, we design capacity of Ublock and Dblock, which is an important indicator to measure the throughput of Healthchain. In the second part, we measure the generation time of the three types of transactions. Furthermore, we measure the generation time of the components of transactions, including encryption and decryption of IoT data and diagnosis, and signature of user and doctor. In the third part, we compare the computation cost and communication cost for user transactions generation of our scheme with that of the traditional scheme. In the experiments we assume that each user has an average of 5 authorized doctors.

A prototype of Healthchain has been implemented to evaluate its efficiency and effectiveness. We simulate the user node with a smart phone, which has a 64-bit 8 core CPU processor, highest 2.45 GHz. The experiment is built on the platform Android 7.1.1. Java programming language is used for prototyping of the IoT transaction and key transaction. Userchain mining nodes and doctor nodes are measured on a 64-bit Windows 7 operating system with Intel(R) Core(TM) i7-4790, 3.60 GHz processor. Userchain and Docchain are written in Python.

A. Capacity of a block

First, we design the structures of Ublock and Dblock. According to the design in Bitcoin [10], the lengths of Preshsh, Index and Merkle root are all set as 32 bytes; Gtime and Nonce are both with the length of 4 bytes. Thus, the key parameters' length setting in the block header is shown in Table I. In addition, in our experiment, we use 1024-bit RSA for asymmetric encryption and signature, 128-bit AES for symmetric encryption, and SHA-256 for hash operation. Therefore, the key parameters' length setting in the block body is indicated in Table II.

The size of tx_{IoT}, tx_{key}, tx_{diag} are 132 Bytes, 1188 Bytes, and 164 Bytes respectively. After considering the merkle tree structure and so on, we can conclude that a Ublock of 1M Bytes can contain either 5349 tx_{IoT} or 837 tx_{key}. A Dblock of 1M Bytes can contain 4599 tx_{diag}. Assuming a Ublock is generated every minute, the throughput can reach 89 tx_{IoT} per second or 13 tx_{key} per second. Assuming a Dblock is generated every minute, then the throughput can reach 76 tx_{diag} per second.

TABLE I
KEY PARAMETERS IN THE BLOCK HEADER

Parameters	Prehash	Index	Gtime	Nonce	merkle root
Length (Bytes)	32	32	4	4	32

TABLE II
KEY PARAMETERS IN THE BLOCK BODY

Parameters	ID	ts	$Signature$	$hash$	$Asymmetric\ encryption$
Length (Bytes)	32	4	32	32	128

B. Processing time of transactions

In this part, we measure the processing time on an Android device and PC respectively. We measure the detail processing time for several major cryptographic operations as shown in Table III.

TABLE III
Processing Time of Transactions

Operation	User	Doctor
SHA-256 (ms)	0.012	7.6×10^{-6}
AES encryption(ms)	0.134	7.3×10^{-5}
RSA encryption (ms)	0.209	4.3×10^{-4}
RSA signing (ms)	3.556	0.021

As shown in Table III, we can find that the processing time of RSA signing is much larger than several other cryptographic operations. Then, we thoroughly test the time the user and the doctor generated the transaction. The time to generate tx_{IoT}, tx_{key} and tx_{diag} by **Algorithm 2**, **Algorithm 3**, and **Algorithm 4** is 3.735 ms, 4.809 ms and 0.021ms respectively. It must also be mentioned that all processing times are the average of 10000 repeated experiments.

C. Comparison of the computation cost and communication cost with traditional scheme

In the third part, we compare the computation cost and communication cost for user transactions generation of our scheme with that of the traditional scheme. In the traditional scheme, the sender encrypts data with a symmetric key. The encrypted data is then sent along with the symmetric key encrypted with the receiver's public key. However, considering that users may update IoT data much more frequently than updating keys. In our scheme, users can update key transactions as needed instead of updating the key each time the IoT transaction is updated. We assume that the user generates a tx_{IoT} every 10 minutes and generates a tx_{key} every 43200

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2923525, IEEE Internet of Things Journal

minutes (about 1 month). Fig. 6 shows the comparison of the computation time overhead of the user transactions generation between our scheme and the traditional scheme. Fig. 7 shows the communication overhead of user transactions generation between our scheme and the traditional scheme.

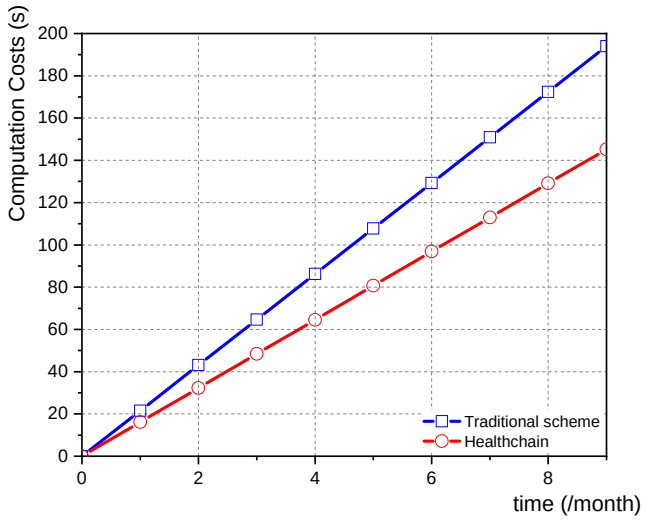


Fig. 6. Computation costs for user transactions generation

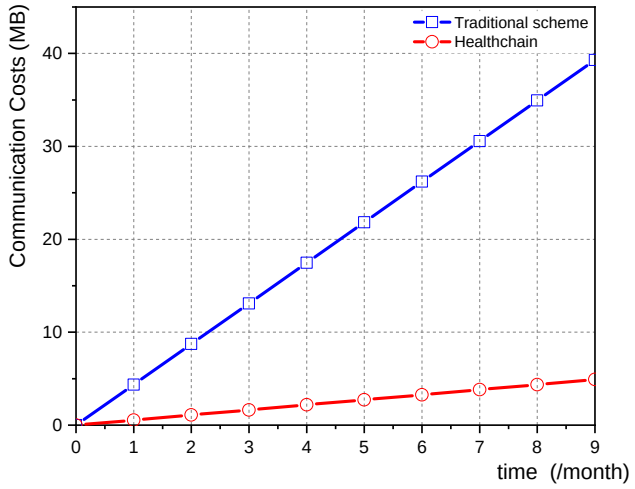


Fig. 7. Communication costs for user transactions generation

As illuminated in Fig. 6 and Fig. 7, both computation cost and communication cost increase as system usage time increases. As shown in Fig. 6, it takes only about 96 seconds for a user to generate user transactions in Healthchain when the system is existing for 6 months, and meanwhile the traditional solution takes about 130 seconds. Compared with the traditional scheme, Healthchain reduces the time for users to generate transactions. As shown in Fig. 7, the size of the transactions generated by users in Healthchain is 3MB when the system is existing for 6 months, and meanwhile the traditional solution generates 26MB. On one hand, it means Healthchain can dramatically decrease the communication overhead for users to send transactions than that in the traditional scheme. On the other hand, it also indicates that our scheme can reduce the size of transactions generated by users and further reduce the storage in blockchain.

VII. Conclusion

In this paper, we proposed a privacy-preserving scheme (Healthchain) for fine-grained access control of large-scale health data based on blockchain. We introduced two blockchains to ensure that both users' health data and doctors' diagnoses cannot be tampered to avoid medical disputes. We decoupled the encrypted data and the corresponding keys to achieve flexible key management. In addition, users can revoke the doctors at any time to ensure the privacy of the user. The security analysis presents that our proposal can meet our expected security requirements. Performance evaluation shows Healthchain is efficient and feasible in practice.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61671420, Youth Innovation Promotion Association CAS under Grant No. 2016394, and the National Key Research and Development Program of China under Grant No. 2016YFB0800301.

References

[1] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[2] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-things-based smart cities: Recent advances and challenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 16–24, 2017.

[3] L. Zhu, C. Zhang, C. Xu, X. Du, R. Xu, K. Sharif, and M. Guizani, "PRIF: A privacy-preserving interest-based forwarding scheme for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2457–2466, 2018.

[4] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multi-subset aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2018.

[5] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for IoT applications in smart homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[6] A. Redondi, M. Chirico, L. Borsani, M. Cesana, and M. Tagliasacchi, "An integrated system based on wireless sensor networks for patient monitoring, localization and tracking," *Ad Hoc Networks*, vol. 11, no. 1, pp. 39–53, 2013.

[7] C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-Healthcare system," *Future Generation Computer Systems*, vol. 79, pp. 16–25, 2018.

[8] Z. Guan, Z. Lv, X. Du, L. Wu, and M. Guizani, "Achieving data utility-privacy tradeoff in Internet of

Medical Things: A machine learning approach," *Future Generation Computer Systems*, 2019.

[9] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Du, and M. Guizani, "LPTD: Achieving lightweight and privacy-preserving truth discovery in CIoT," *Future Generation Computer Systems*, vol. 90, pp. 175–184, 2019.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[11] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[12] N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," in *International Conference on Blockchain*. Springer, 2018, pp. 199–212.

[13] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.

[14] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[15] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4101–4112, 2018.

[16] J. Hua, H. Zhu, F. Wang, X. Liu, R. Lu, H. Li, and Y. Zhang, "CINEMA: Efficient and privacy-preserving online medical primary diagnosis with skyline query," *IEEE Internet of Things Journal*, 2018.

[17] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, N. Gailly, L. Gasser, I. Khoffi, J. Cappos, and B. Ford, "CHAINIAC: Proactive software-update transparency via collectively signed skipchains and verified builds," in *Proceedings of the 26th USENIX Security Symposium*. The Advanced Computing Systems Association, 2017, pp. 1271–1287.

[18] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing Bitcoin security and performance with strong consistency via collective signing," in *Proceedings of the 25th USENIX Security Symposium*. The Advanced Computing Systems Association, 2016, pp. 279–296.

[19] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in Bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 906–917.

[20] T. Ruffing, A. Kate, and D. Schröder, "Liar, Liar, Coins on Fire!: Penalizing equivocation by loss of Bitcoins," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 219–230.

[21] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, and R. Du, "CertChain: Public and efficient certificate audit based on blockchain for TLS connections," in *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM '18)*. IEEE, 2018, pp. 2060–2068.

[22] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, 2018.

[23] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, 2018.

[24] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[25] Z. Shae and J. J. Tsai, "On the design of a blockchain platform for clinical trial and precision medicine," in *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 1972–1980.

[26] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.

[27] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1211–1220, 2017.

[28] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: a new blockchain-based access control framework for the Internet of Things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[29] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, 2018.

[30] A. Al Omar, M. S. Rahman, A. Basu, and S. Kiyomoto, "Medibchain: A blockchain based privacy preserving platform for healthcare data," in *Proceedings of 2017 International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2017, pp. 534–543.

[31] X. Zhang and S. Poslad, "Blockchain support for flexible queries with granular access control to electronic medical records (EMR)," in *Proceedings of 2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

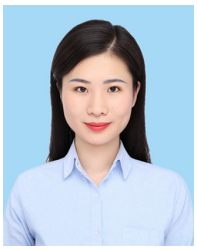
[32] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, no. 10, p. 218, 2016.

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

[34] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.

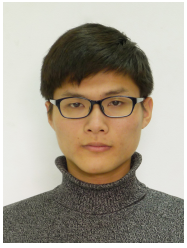
[35] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with china perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.

[36] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. The Advanced Computing Systems Association, 1999, pp. 173–186.



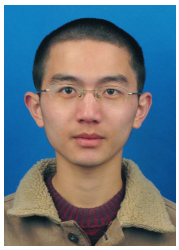
Jie Xu received her B.S. degree from the department of Information Security, University of Science and Technology of China (USTC) in July, 2017. She is currently a graduated student in Communication and Information System from the Department of Electronic Engineering and Information Science(EEIS), USTC. Her research interests include Network security and Cryptography.

Kaiping Xue (M'09-SM'15) received his B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a postdoctoral researcher with Department of Electrical and Computer Engineering, University of Florida. Currently, he is an Associate Professor in the Department of Information Security and Department of EEIS, USTC. His research interests include next-generation Internet, distributed networks and network security. He currently serves as an area editor of Ad Hoc Networks, and associate editors of IEEE Transactions on Network and Service Management, IEEE Access and China Communications. He also serves as a guest editor of IEEE Journal on Selected Areas in Communications (JSAC) and as a lead guest editor of IEEE Communications Magazine. He is the corresponding author of this paper.



Shaohua Li received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2016. He is currently a graduated student in Communication and Information System from the Department of Electronic Engineering and Information Science (EEIS), USTC. His research interests include network security and system security.

Hangyu Tian received his B.S. degree from the department of Information Security, University of Science and Technology of China (USTC) in July, 2018. He is currently a graduated student in Communication and Information System from the Department of Electronic Engineering and Information Science(EEIS), USTC. His research interests include Network security and Cryptography. He currently serves as an area editor of Ad Hoc Networks, and associate editors of IEEE Transactions of Network and Service Management, IEEE Access and China Communications. He also served as a guest editor of IEEE Journal on Selected Areas in Communications (JSAC) and as a lead guest editor of IEEE Communications Magazine. He is the corresponding author of this paper.



Jianan Hong received the B.S. degree from the department of Information Security, University of Science and Technology of China (USTC), in 2012 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2018. Now he is a research Engineer in Huawei Shanghai Research Institute, Shanghai. His research interests include secure cloud computing and mobile network security.

Peilin Hong received her B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1983 and 1986. Currently, she is a Professor and Advisor for Ph.D. candidates in the Department of EEIS, USTC. Her research interests include next-generation Internet, policy control, IP QoS, and information security. She has published 2 books and over 150 academic papers in several journals and conference proceedings.

Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987, the M.E. degree from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2004. Since 1992, he has been a Faculty in the Department of Electronic Engineering and Information Science, USTC, where he is currently a Professor. He is the Executive Director of the Department of Electronic Engineering and Information Science, USTC, and the Director of the Information Processing Center, USTC. He has authored or co-authored more than 130 papers in journals and international conferences. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.