J Med Syst (2018) 42: 140
https://doi.org/10.1007/s10916-018-0995-5

SYSTEMS-LEVEL QUALITY IMPROVEMENT

Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain

Aiqing Zhang[1] · **Xiaodong Lin**[2]

Received: 28 February 2018 / Accepted: 12 June 2018 / Published online: 28 June 2018

Abstract

Electronic health record sharing can help to improve the accuracy of diagnosis, where security and privacy preservation are critical issues in the systems. In recent years, blockchain has been proposed to be a promising solution to achieve personal health information (PHI) sharing with security and privacy preservation due to its advantages of immutability. This work proposes a blockchain-based secure and privacy-preserving PHI sharing (BSPP) scheme for diagnosis improvements in e-Health systems. Firstly, two kinds of blockchains, private blockchain and consortium blockchain, are constructed by devising their data structures, and consensus mechanisms. The private blockchain is responsible for storing the PHI while the consortium blockchain keeps records of the secure indexes of the PHI. In order to achieve data security, access control, privacy preservation and secure search, all the data including the PHI, keywords and the patients' identity are public key encrypted with keyword search. Furthermore, the block generators are required to provide proof of conformance for adding new blocks to the blockchains, which guarantees the system availability. Security analysis demonstrates that the proposed protocol can meet with the security goals. Furthermor, we implement the proposed scheme on JUICE to evaluate the performance.

Keywords Blockchain · Security · Privacy preservation · e-Health · Personal Health Information (PHI)

Introduction

Provision of health services using digital technology has been termed as e-Health. Research trend in the e-Health area is focusing on utilizing the electronic health records for patient monitoring and diagnosis. Usually, a patient may have many healthcare service providers, including primary care physicians, specialists, and therapists [1]. Consequently, health record sharing and exchanging is drawing increasing attentions in industry and research community, where data security and privacy preservation are critical topics in this area.

For a patient, one disease may be caused by or related with another pathema(s). In this case, the precision of the diagnosis is affected by the amount and accuracy of patient's other health information that the doctor gets. The doctor may be provided with some information about the related illness by querying the patient. However, this method is not effective enough to assist the diagnosis because of two reasons: i) If things happened long before, the patient may forget some of the details, such as the medicines or medical examinations he/she had taken, which affects the precision of the diagnosis or treatments he/she had got. ii) The patient cannot professionally describe the diagnosis or treatments due to his/her limited medical knowledge, which affects the judgment of the current doctor. Thus, the current doctor may not be able to deduce correct information for his/her diagnosis.

One promising solution of this problem is to share the patient's health record such that the intended doctor can access the related data for diagnosis improvement. If the patient had visited another doctor in the same hospital or medical institution, and the related health record was generated by

This article is part of the Topical Collection on *Blockchain-based Medical Data Management System: Security and Privacy Challenges and Opportunities*

✉ Aiqing Zhang
aqzhang2006@163.com

Xiaodong Lin
xlin@wlu.ca

[1] College of Physics and Electronic Information, Anhui Normal Univresity, Wuhu, Anhui, 241000 China

[2] Faculty of Science, Wilfrid Laurier University, 75 University Avenue West, Waterloo, ON, N2L 3C5 Canada

this institution, the doctor may access the record directly in the area network under the consent of the patient. However, in practice, the patient may visit different doctors in different medical institutions for different symptoms. Under this circumstance, the doctor will be rejected to provide the data by other institutions without an additional agreement for personal health information (PHI) sharing.

In order to address these issues, cloud-assisted health record sharing has been put forward in the e-Health system as a promising paradigm for health data storage and management [1–6]. These works provide promising solutions to realize PHI sharing among medical institutions in e-Health systems, where security and privacy-preservation are critical concerns. Although all the above works focus on achieving security properties in cloud environments, there remains one challenge: The cloud is expected to be a trusted center to store and manage the data, which is exposed to possible abuse, loss, leakage, or theft if the cloud is under attacks or inadequately supervised. Countermeasures are proposed by employing various cryptographic primitives or other techniques [1–4]. Unfortunately, these security threats remain due to the centralization characteristics of cloud environment.

Blockchain is proposed to be an advantage solution to address the security issues inherited in cloud-based systems because it can maintain a continuously growing list of records, which are distributed and immutable [7–11]. Generally, blockchain is treated as a distributed ledger to store health records for sharing, exchanging or other purposes among stakeholders [12]. In e-Health systems, the patient may visit different medical institutions and each institution manages their own database. PHI sharing systems built on the blockchain technology is expected to achieve secure distribution of health records. Albeit its advantages of immunity and distribution, blockchain based health record sharing system still faces following challenges: i) How to design the consensus mechanism such that blocks are verified without violating the patient's privacy? ii) How to ensure unlinkability while the user's identity is searchable? In other words, unauthorized entity can not link different health records to the same patient. iii) How to guarantee that the authorized doctor is only allowed to access the intended PHI?

In order to address the above issues, we propose to construct a consortium blockchain for secure and privacy-preserving PHI sharing among the hospitals. Each hospital stores the PHI in its private blockchain, which has the advantages of fast transaction, better privacy preservation, low cost, and better security performance. Furthermore, the hospitals are organized to formulate a consortium blockchain, which stores searchable indexes of the PHI. The doctor can search the consortium blockchain for the indexes of interested records and access the original records by visiting the private blockchain of the corresponding hospital.

In summary, the contributions of this work are threefold.

- We propose a framework for blockchain based PHI sharing with security and privacy preservation for diagnosis improvements in e-Health system. Two blockchains are introduced. The private blockchain of the medical service provider stores the patient's original PHI (encrypted for security)[1] while the consortium blockchain keeps records of the secure indexes of the PHI.
- We design core components for the blockchains, including data structure and consensus mechanism. Different block structures are devised for both private blockchain and consortium blockchain. Furthermore, we propose proof of conformance as the consensus mechanism for the blockchains and design an implementation method.
- We propose a secure and privacy-preserving PHI sharing (BSPP) protocol based on the proposed e-Health blockchain. The patient's PHI and the corresponding keywords are encrypted for data security while they are searchable by authorization for diagnosis improvements. Meanwhile, the patients' identities are encrypted for identity privacy preservation. The protocol is carefully designed in such a way that the authorized doctor can search the pseudo identities for the indexes of interested patients. Also, the authorized doctor is only allowed to access the patient's history records while prohibiting from searching for the future records.

The remainder of the paper is organized as follows. An overview on the related work is conducted in "Related Work". Preliminaries are presented in "Preliminaries". Section "System Model" illustrates the system architecture as well as the threat model and design goals of the work. Section "PHI Blockchain Design" designs the PHI blockchain in terms of data structure and consensus mechanism. Based on the proposed blockchain, Section "Blockchain Based PHI Sharing" proposes the PHI sharing protocol in details. Later, Section "Security Analysis" analyses how the protocol achieves the security goals. Section "Implementation and Performance Evaluation" implements the proposed scheme on JUICE and evaluates its performance. Finally, Section "Conclusions" concludes this work.

Related Work

Recent years have witnessed increasing interests of blockchain for security and privacy in e-Health due to its advantages in

[1]The hash value of the encrypted PHI is uploaded to the chain while the original ciphertext is stored in the local computer client.

data management, i.e., immutability and built-in autonomy properties of the blockchain.

Q. Xia et al. [11] proposes a blockchain-based health data sharing framework that sufficiently addresses the access control challenges associated with sensitive data stored in the cloud. The system is based on a permissioned blockchain which allows access to only invited, and hence verified users. Furthermore, in order to provide data provenance, auditing and secured data trailing on medical data, the authors employ smart contracts and an access control mechanism in their another work [13] to effectively track the behavior of the data and revoke access to offending entities on detection of violation of permissions on data .

Yue et al. [14] also proposes a three-layer system: Data usage layer, data management layer and data storage layer. Different from the aforementioned works where cloud is a storage infrastructure, this work proposes that the private blockchain plays the role of cloud. In [15], transactions are used to carry instructions, such as storing, querying and sharing data. The authors combine blockchain and off-blockchain storage to construct a personal data management platform focused on privacy. Kuo et al. [12] review the latest biomedical/health care applications of blockchain technologies. They also discuss the potential challenges and proposed solutions of adopting blockchain technologies in biomedical/health care domains.

Smart contract is constructed in [16] to contain metadata about the record ownership, permissions and data integrity. The contract's state-transition functions carry out policies, enforcing data alternation only by legitimate transactions. Rather than storing the health record in the block, [17] adds addresses of sensors and mobile devices to a healthcare blockchain for pervasive social network (PSN) nodes. Through the addresses stored in the blockchain, a PSN node can visit other nodes in the network and access the health data. This work has the merit of reducing the storage overhead of devices while it did not consider the security of the addresses.

Much like the Bitcoin approach, the block of [18] is a Merkle tree-based structure. The leaf nodes of this tree represent patient record transactions, and describe the addition of a resource to the official patient record. However, transactions do not include the actual record document. Instead, they reference Fast Healthcare Interoperability Resources (FHIR) via Uniform Resource Locators (URLs). Notably, a new consensus algorithm, proof of interoperability, is designed to facilitate data interoperability in this work.

Different from the above works which focus on health data sharing, [19] and [20] focus on different issues. [19] proposes a blockchain platform architecture for clinical trial and precision medicine. This work investigates the design of the blockchain platform starting from the medical domain, particularly the clinical trial and precision medicine. Regarding few studies have been done on key management schemes for blockchain, [20] attempts to develop key negotiation in this area. It uses body sensor networks to design a lightweight backup and efficient recovery scheme for keys of health blockchain. This is pioneering work in key management for blockchain while its performance is greatly influenced by the hardware condition and environment.

The existing works provide diverse frameworks for PHI sharing in e-Health systems with blockchain. Actually, they take the blockchain as an assisted tool for data sharing instead of taking it as a main tool for data storage, data management and data sharing. Additionally, these works do not give a detail solution for a specific application. In this work, We design private and consortium blockchains for PHI sharing without violating the patients' privacy. Also, we propose a PHI secure searching protocol in the system.

Preliminaries

In this section, we give the background and preliminaries required in this paper.

Blockchain

The blockchain is a distributed database that contains an ordered list of records linked together through a chain on blocks [7]. A block usually consists of hash value of previous block, payload, signature of the contributor, and timestamp. The hash value of previous block makes the blockchain immutable to modification. Payload in the block varies with the applications. It can be an address pointer of the original data or the content of the transaction or some other information. Contributor signature and timestamp show the generator and generation time of the block.

In blockchain network, there are two important entities: Miners and verifiers. Miners refer to the nodes who produce new blocks for the blockchain. Different application scenarios may define different nodes as the miners. For example, In the Bitcoin blockchain, the nodes that provide proof of works are provided as the miners to keep records of the transactions. New blocks are accepted only after being verified by verifiers, who are responsible for verifying the validity of new blocks. The processes of generating, verifying and adding new blocks to the blockchain are named mining. In order to guarantee security and reliability of the mining processes, consensus mechanism is critical in blockchain network. It determines who keeps records and how to check the validity of the new block.

As blockchain is originally applied in Bitcoin, transaction is widely used in the network to present the new generated data. In this work, transactions denote the secure indexes of the new emerging health records in the

system. Generally, blockchain can be classified into three categories: Permissionless blockchain (public blockchain), private blockchain and consortium blockchain. In permissionless blockchain, anyone in the world can enter into the system to access the data and send transactions, for example, Bitcoin system. In private blockchain, an organization controls the access right of the system. Consortium blockchain is managed by several organizations. Only the organizations in the system are allowed to access the blockchain.

In relation to our work, private blockchain and consortium blockchain are introduced in the e-Health system to store and manage the user's PHI, which helps improving the diagnosis. Each hospital operates a private blockchain which stores the patient's PHI. The hospitals negotiate to manage a consortium blockchain, which keeps records of the secure indexes for the health records.

Bilinear maps and complexity assumptions

Bilinear maps Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of the same prime order p. A mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is called an admissible bilinear map if it satisfies the following properties:

1. Bilinear: For all $V, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(aV, bQ) = \hat{e}(V, Q)^{ab}$.
2. Symmetric: $\hat{e}(V, Q) = \hat{e}(Q, V)$.
3. Non-degenerate: $\hat{e}(V, Q) \neq 1_{\mathbb{G}_2}$,where $V, Q \neq 1_{\mathbb{G}_1}$.
4. Computable: $\hat{e}$ is efficiently computable.

Complexity assumption Let P be the generator for the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ defined above. The q-BDHI problem is defined as follows: Given the $q+1$-tuple $(P, xP, x^2P, \ldots, x^qP) \in \mathbb{G}_1^{*q+1}$ as input, compute $\hat{e}(P, P)^{1/x} \in \mathbb{G}_2^*$. An algorithm $\mathcal{A}$ has advantage ε in solving q-BDHI in $\mathbb{G}_1$ if

$Pr[\mathcal{A}(P, xP, x^2P, \ldots, x^qP) = \hat{e}(P, P)^{1/x}] \geq \varepsilon$

where the probability is over the random choice of $x \in \mathbb{Z}_p^*$ and random bits of $\mathcal{A}$.

Definition 1 q**-BDHI assumption**. The q-BDHI assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ if all polynomial time algorithms have a negligible advantage in solving the q-BDHI problem.

Keyword Search

Public encryption with keyword search (PEKS) is put forward by Bonth ea al. [21] to achieve search over the encrypted data in asymmetric settings. Usually, a keyword w is extracted from a message m. The keyword is public key encrypted and then searched by using a trapdoor, which is generated by the entity corresponding to the public key. Firstly, we review the definition of PEKS. It consists of four polynomial time randomize algorithms.

KeyGen(λ): Input a security parameter λ and output a public/private key (pk, sk).

PEKS(pk_i, w): Given entity i's public key and a keyword w, it produces a searchable encryption c_w for w.

Trapdoor(sk_i, w'): Take i's private key sk_i and a keyword w' as input, output a trapdoor $T_{w'}$.

Test($pk_i, c_w, T_{w'}$): Given i's public key pk_i, a searchable encryption c_w and a trapdoor $T_{w'}$, output "yes" if $w = w'$ and "no" otherwise.

Afterwards, various keyword search algorithms are proposed to provide diverse search functions by combining PEKS with other cryptographic primitives. In order to enhance the search security, PEKS schemes with designed tester is proposed in [22, 23]. Combined with proxy re-encryption [24, 25], the keyword search mechanisms allow a delegatee to search for the interested keywords from the delegator's data. Keyword search with oblivious transfer is introduced in [26] to address the user privacy issue. The oblivious keyword search prohibits the data supplier from knowing the chosen keywords and the corresponding ciphertext. Regarding some applications require more than one keyword to be searched for, pubic key encryption with conjunctive field keyword search is presented in the works [27–29]. The above PEKS schemes may be mixed together to achieve the required security objectives. [3] proposes a conjunctive keyword search with designed tester and proxy re-encryption function.

In this work, based on the keyword search algorithm [30], we propose a secure and privacy preserving keyword search protocol for blockchain-based health record system. By introducing a polynomial constructed on the keyword set, the proposed protocol is able to provide proof of conformance for the blockchain, which works as the consensus mechanism.

System Model

In this section, we present the system architecture for the blockchain based health record sharing system. And then, we illustrate the threat model and design goals.

System Architecture

Assume that several hospitals in a city agree to form a league to share their patients' health record. In order to enhance security of the data, two kinds of blockchains are constructed in the system: Private blockchain of each hospital and consortium blockchain of the hospitals in the coalition. The private blockchain stores the original PHI of the patients that had visited the hospital while

the consortium blockchain stores keywords of the PHI generated by all the hospitals in the alliance. Basically, the framework of the proposed secure and privacy-preserving PHI sharing system is shown in Fig. 1. There are three entities in the system: System manager, medical service providers (hospitals), and users (patients).

System manager. System manager charges the whole system. All the users and doctors are required to register to the system manage. It generates system parameters and keeps a public key tree for the doctors and users. It also generates consensus vector **a** (it will be described in "PHI Blockchain Design") for the consortium blockchain.

Medical service providers. Medical service providers are medical institutions which provide medical services for the users. In this work, we refer to hospitals for simplification of expression. Usually, each hospital poses a server and many computer clients. Each computer client is operated by a doctor to record his/her patients' health information. Then the clients generate blocks for the patients' health records and broadcast them to the private blockchain of the hospital. Moreover, the selected computer client is responsible for verifying the new coming blocks. The server is responsible for the registrations of the users by keeping a register table for the users and doctors. It also takes the responsibility of collecting new blocks in the private blockchain at a predefined interval and formulating new blocks for the consortium blockchain. Also, the selected server should verify new blocks for the consortium blockchain. Notably, the server authenticates the doctors outside the private blockchain for their accessing of the user's PHI in the private blockchain.

Users. Users refer to the patients who visit hospitals for services. They must register to the server of the hospital before seeing the doctor. Each patient will get a token from the server after registration. The patient should keep the token secret and show it to the doctor when coming to the doctor. The beacon works as an evidence for the interaction of the doctor and the patient, which authorizes the doctor to generate PHI for the patient. The PHI is stored in the private blockchains of the hospital. As one patient may visit different doctors in different hospitals for different illnesses, his/her PHI is stored in the private blockchains of the corresponding hospitals. Furthermore, the hospitals send searchable keywords of blocks in their private blockchains to the consortium blockchain. In this way, PHI is stored with distribution and immutability, and they are searchable. In this work, "user" and "patient" are interchanged.

Usually, current hospitals own servers. They can also provide computer clients for the doctors. Consequently, private blockchains and consortium blockchain can be constructed based on the existing infrastructures of the hospitals without additional instruments. Notably, softwares are required to be installed in the servers and computers for establishing private blockchains and consortium blockchain.

The notations used throughout the paper are listed in Table 1.

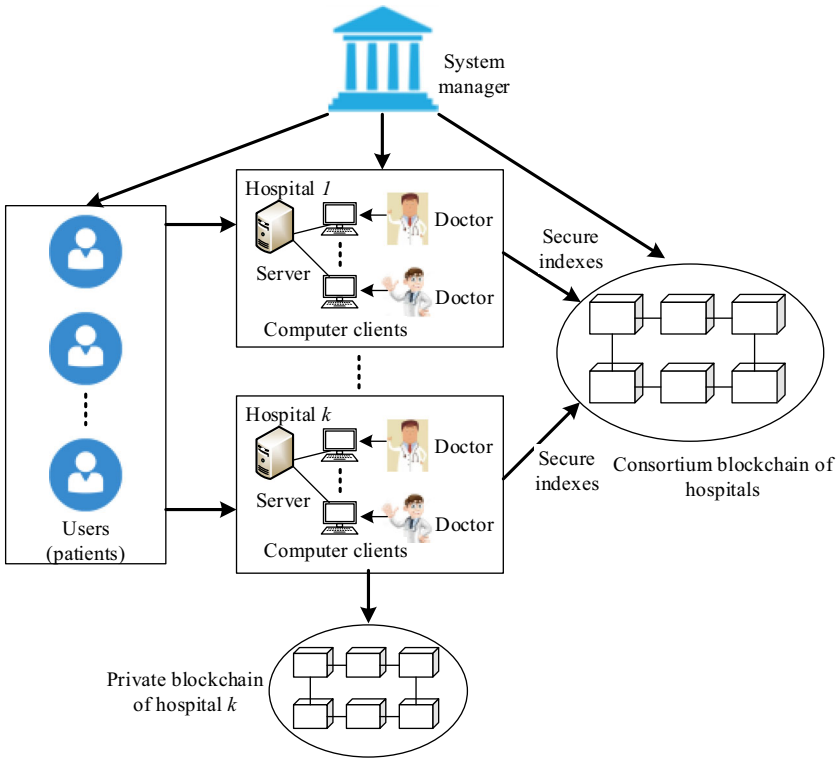

Fig. 1 System architecture

Table 1 Notations and description

Notation	Description	Notation	Description
$\mathcal{W}$	Keyword set	ID_i	User i's identity
$\mathcal{U}$	User set	ID_j	Doctor j's identity
$\mathcal{D}$	Doctor set	ID_b	The identity of a block
$\mathcal{H}$	Hospital set	T_w	Keyword searching trapdoor
Tx_i	Secure index	T_d	Identity searching trapdoor
σ_j	j's signature	pk_i, sk_i	Public key and private key of i
d_i	i's pseudo identity	c_e, η	Evidence for proof of conformance

Threat Model and Design Goals

The hospital servers and computer clients are deemed as semi-trusted. They are honest to perform the protocol but curious to access or deduce the user's health information without authorization. The outsider attackers can eavesdrop the transmissions in the public channel, such as secure indexes, encrypted PHI, and trapdoors. The computer clients are not allowed to collude with the server to infer the real identity of the user.

Based on the above threat models, we aim to achieve the following goals:

Data security and access control. As PHI is privacy sensitive, it is critical to achieve data security, including data confidentiality and integrity, data auditing, and access control. Usually, data confidentiality and integrity are guarantee by encryption and signature. Data auditability and access control should be achieved to ensure that all the data access activities are monitored under the data owners (patients) and the data generators (hospitals). They can be achieved through identification, authentication and authorization by using cryptographic primitives. In this work, data security can be enhanced by combination of private blockchain and consortium blockchain, which store the PHI and the corresponding keywords respectively.

Privacy-preservation. Albeit privacy preservation can be achieved partly through data confidentiality and access control, user's identity leakages some privacy-sensitive information in e-Health systems. Consequently, it is vital important to keep the user's identity information secret. Generally, anonymity and unlinkability are required to realize identity privacy. Here, unlinkability means that the eavesdroppers are not able to judge whether two or more flows of PHI come from the same source.

Secure search. In this system, the doctor is authorized by a patient to search for his/her history PHI in order to improve diagnosis. During this process, only the authorized doctor is allowed to access the interested content. The eavesdroppers are not able to guess the keywords. Moreover, as the doctor has to search the pseudo identities for the intended patient, the eavesdroppers are also not allowed to deduce the real identities of the users.

Time controlled revocation. After getting keyword searching trapdoor and identity searching trapdoor from the patient, the doctor is authorized to access the user's history record. However, he/she should not be able to access the user's future health records using the same searching trapdoors. In other words, access right should be time controlled.

System availability. In this work, system availability includes two aspects: i) The secure keywords stored in the consortium blockchain should be selected from the allowable keyword set such that they are searchable. ii) The pseudo identities for the same patients should be able to be figured out by the authorized doctors to search for their PHI keywords.

PHI Blockchain Design

In this section, we design the blockchain for the PHI sharing system from the aspects of data structure, and consensus mechanism.

Data Structure

As private blockchain and consortium blockchain store different contents in the system, the blocks of the two kinds of blockchains have different structures. The data structure of the private blockchain is shown in Fig. 2. It consists of block header, payload, signature of the contributor, and timestamp. Block header concludes three components: Block ID, block size, and hash value of previous block. Payload is composed by four parts: Identity of the PHI generator (doctor), pseudo identity of the PHI owner (patient), encrypted PHI hash and its keyword. Notably, all the PHI related information is stored in the format of ciphertext for privacy preservation of the patient. Specifically, the block stores the pseudo identity of the data owner, which is derived from the true identity; the PHI

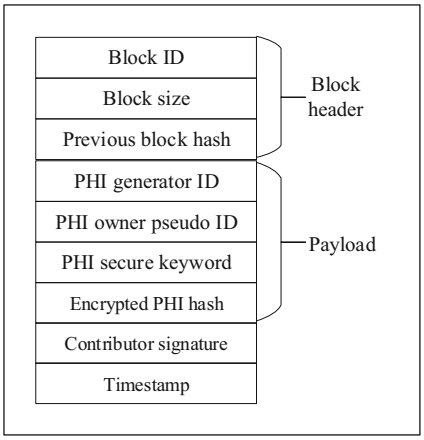

Fig. 2 Structure of a block in the private blockchain of a hospital

and its keyword are encrypted in the block. Contributor signature helps to track the generator (doctor) of the block. Timestamp shows the generation time of the block.

Similarly, the block in the consortium blockchain is composed of block header, payload, signature of the contributor, and timestamp, as shown in Fig. 3. The block generator (hospital server) creates a block at an interval, during which the server collects the health records generated in the time period. The block only stores secure indexes in the payload on the consortium blockchain instead of keeping the original PHI. A secure index is composed by $l(l \geqslant 1)$ transactions $Tx_1, Tx_2, \ldots, Tx_l$. Each transaction records a secure index for a patient's PHI, including three items: Block ID, PHI owner pseudo ID and PHI keyword.

Consensus Mechanism

Consensus mechanism is the core technology for blockchain as it determines whether the new block is validated and who keeps the record. Thus, it influences the security and reliability of the whole system. We propose proof of conformance as the network consensus mechanism in the system for both private blockchain and consortium blockchain.

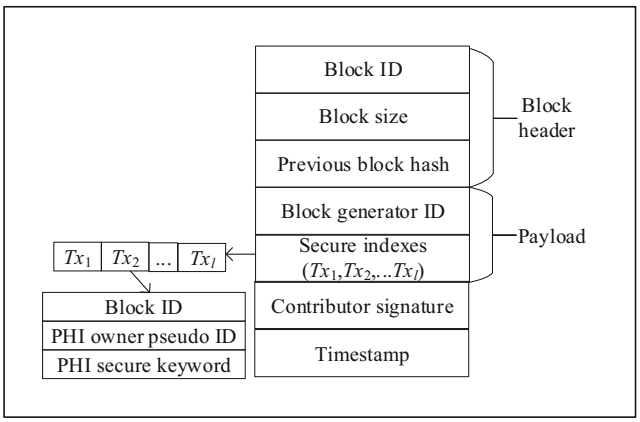

Fig. 3 Structure of a block in the consortium blockchain

Remark 1 As discovered in [31], after 2/3 of all the participates verifying the replicas in a distributed system, the system is fault tolerance. Consequently, we define that a new block is acceptable if it is verified by 2/3 of all the nodes in the blockchain.

Consensus mechanism for private blockchain. When a user (patient) goes to a hospital searching for medical service, he/she will register to the hospital server and select a doctor as his/her service provider. After interaction between the patient and the doctor, the patient's PHI will be generated by the doctor. The doctor encrypts the PHI with the user's public key and generates a pseudo identity d_i for the user. The d_i is appended with the encrypted PHI to label the owner of the PHI. A new transaction for this PHI is sent to the private blockchain of the hospital. Then, the verifiers verify the block by checking whether the PHI is generated by the authorized doctor. This consensus mechanism is defined as proof of conformance, i.e., only the block generated by the doctor, whom the user with pseudo identity d_i has visited, is validated. This consensus mechanism can be achieved by generating a secure token for the user after he/she registers to the hospital. When arriving at the doctor, the user shows the doctor the token. Then the doctor can computer a pseudo identity for the user by using the secure token.

After receiving a new transaction generated by this client, the other clients check whether the doctor is authorized by the patient to generate the data. If more than 2/3 of clients verify the new transaction, it is accepted as a new validate block in the private blockchain. Section "Protocol description" presents this consensus scheme in details.

Consensus mechanism for consortium blockchain. Similar to the private blockchain, proof of conformance also plays the role of consensus mechanism in the consortium blockchain in the system. However, it requires conformance of different contents. Specifically, the consortium blockchain provides keyword search for the users, thus the keywords stored in the blockchain should be interoperability. In the system, as the keywords describe the symptoms or diagnosis of the patients, they conform to standard medical statements, such as FHIR profiles [32, 33][2]. Usually, the

[2]Fast Healthcare Interoperability Resources is a standard describing data formats and elements for exchanging electronic health records. Its goals is to facilitate interoperation between legacy health care systems. This resource makes it easy to provide health care information to health care providers and individuals on a wide variety of devices from computers to tablets to cell phones. It also allows third-party application developers to provide medical applications which can be easily integrated into existing systems.

keywords are constrained in a predefined set for the users to search in the blockchain. The consortium blockchain reaches network consensus: Verify that keywords of the secure indexes in the incoming blocks are selected from $\mathcal{W}$, where $\mathcal{W}$ denotes the predefined keyword set.

In order to implement proof of conformance in the consortium blockchain, the system constructs a polynomial based on the keywords. We assume that $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$, where n is the size of $\mathcal{W}$. The polynomial is constructed as follows:

Compute $H_1(w_1), H_1(w_2), \ldots, H_1(w_n)$ and construct a polynomial $f(x)$ with order n, satisfying $f(H_1(w_i)) = 0, i \in \{1, 2, \ldots, n\}$. The polynomial can be denoted as

$$f(x) = (x - H_1(w_1))(x - H_1(w_2)) \ldots (x - H_1(w_n)). \quad (1)$$

Equation 1 can be reorganized as

$$f(x) = x^n + b_{n-1}x^{n-1} + \ldots + b_1 x + b_0,$$

where $[1, b_{n-1}, b_{n-2}, \ldots, b_0] \triangleq \mathbf{b}$ are the coefficients of the polynomial. The equation $f(x) = 0$ is transformed into

$$x^n + b_{n-1}x^{n-1} + \ldots + b_1 x = -b_0. \quad (2)$$

Subdivide $-b_0$ at both sides of Eq. 2, then it turns to be

$$\frac{-1}{b_0}x^n + \frac{-b_{n-1}}{b_0}x^{n-1} + \ldots + \frac{-b_1}{b_0}x = 1. \quad (3)$$

Let $a_n = \frac{-1}{b_0}$, $a_{n-1} = \frac{-b_{n-1}}{b_0}, \ldots, a_1 = \frac{-b_1}{b_0}$. Construct a new polynomial

$$g(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x. \quad (4)$$

It can be deduced that $g(H_1(w_i)) = 1$, where $w_i \in \mathcal{W}$. Define vector $\mathbf{a} = [a_1, a_2, \ldots, a_{n-1}, a_n]$ and vector $\mathbf{h}_i = [H_1(w_i), (H_1(w_i))^2, \ldots, (H_1(w_i))^{n-1}, (H_1(w_i))^n]$. Then, the inner product of vector $\mathbf{a}$ and vector $\mathbf{h}_i$, $\mathbf{a} \cdot \mathbf{h}_i = 1$. The vector $\mathbf{a}$, named consensus vector, helps to check the validity of the secure indexes in the new appearing blocks in the consortium blockchain.

If more than 2/3 of hospital servers verify the new transactions, it is accepted as a new validate block in the consortium blockchian. Section "Protocol description" designs the detailed consensus mechanism for consortium blockchain.

Blockchain Based PHI Sharing

In this section, we firstly present an overview of the proposed protocol for a comprehensive understanding of the scheme. Then, we describe the protocol in details.

Overview of the Scheme

Fig. 4 describes the procedure of the proposed protocol. The protocol can be divided into three layers: Data generation layer, data storage layer, and data service layer.

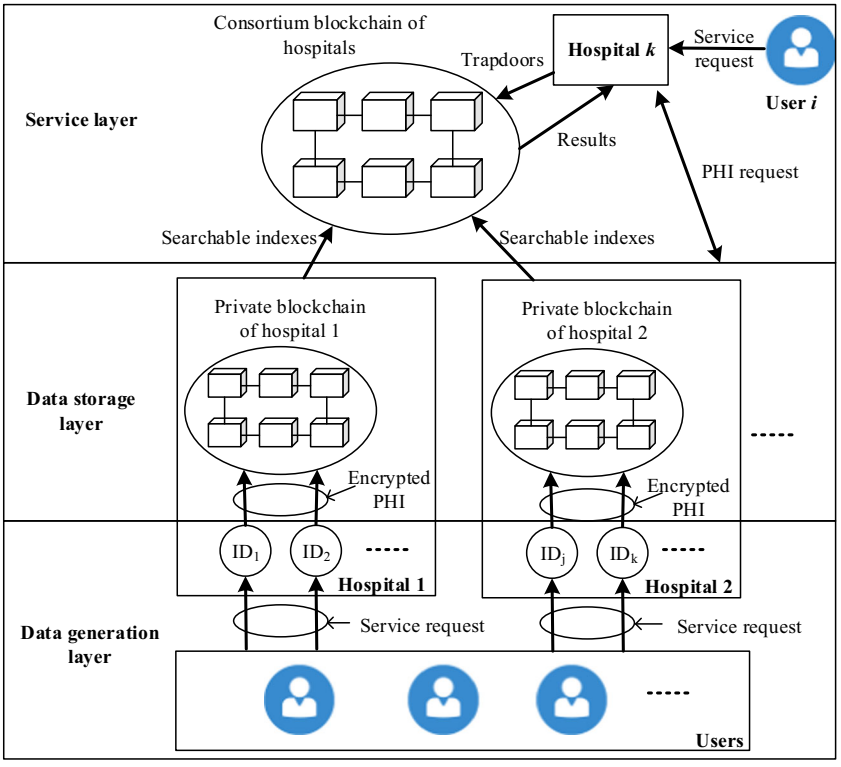

Fig. 4 The proposed protocol

Without loss of generality, we assume that a patient $i \in \mathcal{U}$ registers to hospital $k \in \mathcal{H}$ to see the doctor $j \in \mathcal{D}$, the server of hospital k generates an evidence β for the user and sends it to the user. Meanwhile, it stores $\mu = H_1(\beta)$ in the server register table for doctor j. Here, β works as an authorization for the doctor to generate PHI for the user i. After the patient i physically visits the doctor, he/she gives β to the doctor as a consent for generating his/her PHI and accessing his/her history health record. Suppose that the doctor generates health record m for user i by the interaction. For safely storing the data with interoperability, the doctor extracts a keyword $w \in \mathcal{W}$ for the PHI. Then, it encrypts m and w with the user's public key pk_i. The ciphertext $c = [c_{i0}, c_{i1}, c_{i2}]$ is composed by three components: c_{i0} is the ciphertext of PHI m, which is stored in the private blockchain; c_{i1} is the ciphertext of the keyword w, which is stored in the private and consortium blockchain for searching purposes; c_{i2} is a promise for proof of conformance sent to the consortium blockchain.

Furthermore, the doctor computes a pseudo identity d_i for the user. The pseudo identity should satisfy the following requirements: i) It can not be linked with the user's real identity by the other entities without the user's authorization for privacy preservation. Meanwhile, a doctor can find out the real identity under the authorization of the user. ii) It is different from the user's other pseudo identities generated by other doctors to achieve unlinkability. In order to achieve these goals, the doctor encrypts the user's identity by using searchable encryption. The ciphertext works as the pseudo identity. Finally, a new transaction is formulated by the client in the formate of TX in Table 2 for the private blockchain, where $hash$ denotes the hash value of the previous block, σ_j denotes the doctor's signature, and t denotes the timestamp. Notably, for proof of conformance in the private blockchain of the hospital, the client outputs a token η, which is a function of β and her/his private key sk_j. The validity of the block is checked by using μ and the public key pk_j.

The server of a hospital k will collect all the new blocks and send the secure indexes to the consortium blockchain. In this work, we assume that after 8 new blocks are generated[3] in the private blockchian, the server will extract block identity, user pseudo identity, and secure keywords from each block to formulate a new transaction for the consortium blockchain, as shown in transaction TX of Table 3. When a new transaction arrives, the verifiers in the consortium blockchain extract $Tx_i, i \in \{1, 2, \ldots, 8\}$ and checks its validity. The detailed steps of verification will be described in subsection 2. If all the secure indexes are validated, the new blocks are added to the consortium blockchain.

[3] To avoid the case that less than 8 new blocks are generated in a long period, a time interval can be predefined in the system.

Table 2 Block generated by doctor j for user i in private blockchain

Block header			Transaction TX				Timestamp
Block identity	Block size	Previous block hash	Data generator ID	User pseudo identity	Secure keywords	Encrypted PHI hash	Contributor signature	
ID_b	$size$	$hash$	ID_j	d_i	(c_{i1}, c_{i2})	$hash(c_{i0})$	σ_j	t
32 Bytes	4 Bytes	32 Bytes	9 Bytes	$(3\|G_1\| + \|Q\|)$ Bytes	$((n+3)\|G_1\| + \|G_2\| + \|Q\|)$ Bytes	32 Bytes	9 Bytes	9 Bytes

Table 3 Block generated by hospital k for consortium blockchain

Block header			Transaction TX			Timestamp
Block identity	Block size	Previous block hash	Data generator ID	Secure indexes	Contributor signature	
ID_b	$size$	$hash$	ID_k	$(Tx_1, Tx_2, \ldots, Tx_l)$	σ_k	t
32 Bytes	4 Bytes	32 Bytes	9 Bytes	$8(9+6\|G_1\|+2\|Q\|)$ Bytes	9 Bytes	9 Bytes

Let Tx_i denote the secure index for the data generated by doctor j for user i. Then, $Tx_i = (ID_b, d_i, c_{i1})$. A new block generated by the hospital k for consortium blockchain is shown in Table 3.

If the doctor j thinks it is necessary to search for the user i's history health record related with keyword w' before generating his/her PHI m, it comes to the second stage: Data search. After achieving an agreement, the user i sends the doctor an identity searching trapdoor T_d and a keyword searching trapdoor $T_{w'}$. The T_d helps the doctor to find out all the secure indexes of the user's history records in the consortium blockchain. Furthermore, the doctor can find out the intended secure index Tx_i which concludes keyword w' by using $T_{w'}$.

In order to recover the PHI related with keyword w', the doctor parses $Tx_i = (ID_b, d_i, c_{i1})$. By checking ID_b, the doctor can access the corresponding private blockchain to extract the ciphertext c_{i0}. The doctor obtains the user's original PHI m through decrypting c_{i0}. The following subsection describes the detailed constructions.

Protocol description

The proposed scheme is composed by three phases: System setup, data generation and storage, data search and access.

Phase 1: System setup.

In this step, the system generates the system parameter GP by implementing the function $GlobalSetup(\lambda)$, where λ is the security parameter. All the users $i \in \mathcal{U}$ and doctors $j \in \mathcal{D}$ generate their private keys and public keys by operating the function $KeyGen(GP)$.

- $GlobalSetup(\lambda)$: Choose two bilinear groups $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order p and a bilinear map $\hat{e}$. Pick two different generator P_1, P_2 of $\mathbb{G}_1$ and set $g = e(P_1, P_1)$, $h = e(P_1, P_2)$. Pick seven hash functions $H_0 : \mathbb{G}_2 \rightarrow \{0,1\}^*$, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2 : \mathbb{G}_1 \rightarrow \mathcal{M}$ (the message space), $H_3 : \mathcal{M} \times \mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$, $H_4 : \mathbb{G}_1 \times \mathbb{Z}_p^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$, $H_5 : \{0,1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$, $H_6 : \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$. The system parameter $GP = (P_1, P_2, \hat{e}, H_0, H_1, H_2, H_3, H_4, H_5, H_6)$.
- $KeyGen_i(GP)$: Each user $i \in \mathcal{U}$ randomly chooses numbers $(y_{i1}, y_{i2}, y_{i3}) \in \mathbb{Z}_p^*$ as its private key sk_i. It outputs the public key $pk_i = (Y_{i1} = y_{i1}P_1, Y_{i2} = y_{i2}P_1, Y_{i2} = y_{i2}P_1)$.
- $KeyGen_j(GP)$: Each doctor $j \in \mathcal{D}$ randomly chooses numbers $y_j \in \mathbb{Z}_p^*$ as its private key sk_j. It outputs the public key $pk_j = (Y_j = y_j P_2)$.

Phase 2: Data generation and storage.

When user i registers to hospital k, the server of the hospital randomly selects $\beta \in \{0,1\}^*$ and sends β to the

user securely. Meanwhile, the server chooses a doctor j for the user[4]. The server computes $\mu = H_1(\beta)$ and stores it in the system with the doctor. As the user comes to the doctor j, he/she will show the doctor β, which works as an evidence for the user's authorization to the doctor for generating his/her PHI.

After interaction with patient i, the doctor j generates health record $m \in \{0,1\}^*$ as the patient's PHI and selects the keyword $w \in \{0,1\}^*$ from the standard keyword set. The doctor encrypts m and w with the user's public key pk_i by implementing the following operations.

- Randomly chooses $r_1 \in \mathbb{Z}_p^*$ and computes $B = r_1 Y_{i2}$, $r_0 = H_3(m, B)$, $A = r_0(Y_{i1} + H_1(w)P_1) + r_1 H_1(w)P_1$, $E = r_0 Y_{i3}$, $F = H_4(h^{r_0}, A, B)$.
- Computes $J = g^{r_0(y_{i1}+H_1(w))}$ and vector $\mathbf{X} = [X_1, X_2, \ldots, X_n]$, where $X_1 = r_1 H_1(w)P_1$, $X_2 = r_1(H_1(w))^2 P_1, \ldots, X_n = r_1(H_1(w))^n P_1$.
- Computes $c_{i0} = H_2(h^{r_0}) \oplus m$.

The output of the encryption algorithm is $c = (c_{i0}, c_{i1}, c_{i2})$, where $c_{i1} = (A, B, E, F)$ and $c_{i2} = (J, \mathbf{X})$. Here, (A, B, E, F) is the searchable keyword ciphertext and $c_e = (A, J, \mathbf{X})$ is the evidence for proof of conformance in the consortium blockchain. Notably, A can resist keyword cheating as it exists both in the keyword ciphertext c_{i1} and the evidence c_e. The ciphertext c_{i0} is stored in the hospital and the blockchain stores the hash value of c_{i0}.

Moreover, the doctor generates a pseudo identity d_i for the user by encrypting ID_i ($ID_i \in \{0,1\}^*$) with ID_i as the keyword. Similar to the encryption for m and w, the doctor performs the following operations:

- Randomly chooses $r_1 \in \mathbb{Z}_p^*$ and computes $B_d = r_1 Y_{i2}$, $r_0 = H_5(ID_i, B_d)$, $A_d = r_0(Y_{i1} + H_1(ID_i)P_1) + r_1 P_1$.
- Computes $E_d = r_0 Y_{i3}$, $F_d = H_4(h^{r_0}, A_d, B_d)$.

Then $d_i = (A_d, B_d, E_d, F_d)$. Additionally, in order to provide a proof of conformance in the private blockchain, the doctor generates a proof $\eta = (\alpha, \beta')$ based on β and his private key sk_j, where α and β' are computed by doctor j as follows:

- Randomly chooses $r \in \mathbb{Z}_p^*$, computes $\alpha = \frac{r}{y_j + H_1(\beta)}$.
- Computes $\beta' = H_0(rP_2) \oplus \beta$.

Finally, the doctor formulates the transaction TX of Table 2 and broadcasts it to the private blockchain of hospital k. Upon receiving a new transaction, the verifiers of the private blockchain verify it as follows:

- Extracts ID_j, $\eta = (\alpha, \beta')$ from the block and searches μ for ID_j in the system.
- Computes $\beta* = H_0(\alpha(Y_j + \mu P_2)) \oplus \beta'$ and checks whether $H_1(\beta*) = \mu$.

If the equation holds, the transaction is validate. The verifier broadcasts a validation confirm message. After receiving $\lfloor \frac{2}{3} n_p \rfloor$ validation confirm messages, the new transaction is accepted. Here, n_p denotes the amount of nodes in the private blockchain. It is structured as Table 2 and added to the blockchain.

Correctness.

$$\begin{aligned} H_1(\beta*) &= H_1(H_0(\alpha(Y_j + \mu P_2)) \oplus \beta') \\ &= H_1(H_0(\tfrac{r}{y_j + H_1(\beta)}(y_j P_2 + H_1(\beta)P_2)) \oplus \beta') \\ &= H_1(H_0(rP_2) \oplus H_0(rP_2) \oplus \beta) \\ &= H_1(\beta) \\ &= \mu \end{aligned}$$

In each private blockchain, the system server extracts the block identity, user pseudo identity and secure keyword of each new blocks. Let $Tx_i = (ID_b, d_i, c_{i1})$, $i \in \{1, 2, \ldots, 8\}$. The server composes secure indexes of all the new emerging 8 blocks. These secure indexes are formulated as a new transaction in the format of Table 3. The server broadcasts the new transactions with the ciphertext c_{i2}, $i \in \{1, 2, \ldots, 8\}$ to the consortium blockchain. When a new transaction arrives, the verifiers of consortium blockchain verify each secure index Tx_i, $i \in \{1, 2, \ldots, 8\}$ in the block as follows:

- Parses A from c_{i1} and $c_{i2} = (J, \mathbf{X})$. Checks whether $\hat{e}(A, P_1) = \hat{e}(X_1, P_1) \cdot J$.
- Checks whether $\hat{e}(\mathbf{a} \cdot \mathbf{X}, X_2) = \hat{e}(X_1, X_1)$.

If both equations hold, the block is accepted. Otherwise, it is aborted.

Correctness.

$$\begin{aligned} \hat{e}(A, P_1) &= \hat{e}(r_0(Y_{i1} + H_1(w)P_1) + r_1 H_1(w)P_1, P_1) \\ &= \hat{e}(r_1 H_1(w)P_1, P_1)\hat{e}(r_0(Y_{i1} + H_1(w)P_1), P_1) \\ &= \hat{e}(X_1, P_1)\hat{e}(P_1, P_1)^{r_0(y_{i1}+H_1(w))} \\ &= \hat{e}(X_1, P_1) \cdot J \end{aligned}$$

$$\begin{aligned} \hat{e}(\mathbf{a} \cdot \mathbf{X}, X_2) &= \hat{e}(a_1 r_1 H_1(w)P_1 + a_2 r_1 (H_1(w))^2 P_1 + \ldots \\ &\quad + a_n r_1 (H_1(w))^n P_1, r_1 (H_1(w))^2 P_1) \\ &= \hat{e}(r_1 P_1, r_1 (H_1(w))^2 P_1) \\ &= \hat{e}(r_1 H_1(w)P_1, r_1 H_1(w)P_1) \\ &= \hat{e}(X_1, X_1) \end{aligned}$$

Phase 3: Data search and access.

During the interaction processes of a doctor and a user, the doctor may find it is necessary to access the user's history record for more precise diagnosis. In this case, the doctor (assume doctor j) will ask for an identity searching trapdoor and a keyword w searching trapdoor from the user. The user (assume user i) generates the identity searching trapdoor T_d and keyword trapdoor T_w for the doctor.

[4]The doctor can also be chosen by the user in practical applications.

Remark 2 Generally, the doctor is only authorized to access the user's history record. He/she is prohibited to access the future record without authorization. Consequently, the searching doors should be designed in such a way that they only work when the doctor searches the data within the allowable time window. In order to achieve this security goal, we define an enable function $f(x)$ which has the follow properties:

$$f(x_1)/f(x_2) = \begin{cases} 1 & x_1 \geq x_2 \\ 0 & x_1 < x_2 \end{cases} \quad (5)$$

In the system, the current timestamp t_s can be expressed as an integer. For example, the current time "2:00 PM, Dec. 27, 2017" can be denoted as "$t_s = 2017122714$". The history record refers to the record generated before "2:00 PM, Dec. 27, 2017", i.e., the data generation time $t_0 \leq t_s$.

Assume that the current timestamp is t_s. The user performs the following operations to generate $T_d = (T_1^d, T_2^d, T_3^d)$ and $T_w = (T_1, T_2, T_3)$:

- Randomly chooses $r_d \in \mathbb{Z}_p^*$, computes $T_1^d = Y_j/(y_{i1} + H_1(ID_i) + y_{i3}H_6(\beta, r_d))$, $T_2^d = T_1/y_{i2}$, $T_3^d = r_d f(t_s)$.
- Randomly chooses $r \in \mathbb{Z}_p^*$, computes $T_1 = Y_j/(y_{i1} + H_1(w) + y_{i3}H_6(\beta, r))$, $T_2 = T_1/y_{i2}$, $T_3 = rf(t_s)$.

After getting the searching trapdoors, the doctor first searches the pseudo identity in the secure indexes of consortium blockchain to find out the indexes for user i. Specifically, the doctor extracts secure indexes $(Tx_1, Tx_2, \ldots, Tx_8)$ and the timestamp t_0 of the blocks. For each $Tx_i = (ID_b, d_i, c_{i1}), i \in \{1, 2, \ldots, 8\}$, parse d_i as $d_i = (A_d, B_d, E_d, F_d)$. The doctor j checks whether d_i is the pseudo identity of ID_i as follows:

- Computes $U_1 = \hat{e}(A_d + H_6(\beta, T_3^d/f(t_0))E_d, T_1^d)^{1/y_j}$, $U_2 = \hat{e}(B_d, T_2^d)^{1/y_j}$.
- Computes $V_d = U_1/U_2$, checks whether $H_4(V_d, A_d, B_d) = F_d$.

Correctness.

$$\begin{aligned} V_d &= U_1/U_2 \\ &= \hat{e}(r_0(Y_{i1} + H_1(ID_i)P_1) + r_1P_1 + H_6(\beta, \tfrac{r^d f(t_3)}{f(t_0)})r_0Y_{i3}, \\ &\quad \tfrac{Y_j}{y_{i1} + H_1(ID_i) + y_{i3}H_6(\beta, r_d)})^{1/y_j} \\ &\quad /\hat{e}(r_1Y_{i2}, Y_j/((y_{i1} + H_1(ID_i) + y_{i3}H_6(\beta, r_d))y_{i2}))^{1/y_j} \\ &= \hat{e}(r_0P_1, Y_j)^{1/y_j}(when t_0 < t_s) \\ &= h^{r_0} \end{aligned} \quad (6)$$

Remark 3 A dishonest doctor may input false t_0, i.e. $t_0' < t_0$ to achieve $t_0 < t_s$ such that $H_6(\beta, r_d f(t_s)/f(t_0)) = H_6(\beta, r_d)$. In this way, the doctor may use the trapdoor to access the patient's future health record. In order to address this issue, we propose that the value of $f(t_0)$ should be generated by the block sender.

If the equation holds, Tx_i is the user i's secure index, the doctor goes on implementing the following operations to check whether the index is for keyword w. Otherwise, it aborts.

- Parses $c_{i1} = (A, B, E, F)$ and computes $U_1 = \hat{e}(A + H_6(\beta, T_3/f(t_0))E, T_1)^{1/y_j}$, $U_2 = \hat{e}(B, T_2)^{1/(y_j H_1(w))}$.
- Computes $V = U_1/U_2$, checks whether $H_4(V, A, B) = F$.

Correctness. The proof of the correctness is similar with Eq. 6.

If the equation does not hold, c_{i1} is not keyword w's secure index. The doctor aborts. Otherwise, the doctor temps to access the original PHI by accessing the private block which generates this secure index. In specific, the doctor checks the block identity ID_b and finds out the corresponding private blockchain, denoted hospital k's private blockchain. Then, the doctor logins in the server of hospital k for accessing the block ID_b. The server authenticates the doctor and locates the block ID_b in the blockchain. It parses $hash(c_{i0})$ from the block, and sends the ciphertext c_{i0} to the requesting doctor. Then, The original PHI m can be obtained by the doctor through decrypting c_{i0} as follows:

- Computes $m' = c_{i0} \oplus H_2(V)$, and $r_0' = H_3(m', B)$. Checks whether $h^{r_0'} = V$.

If the equation holds, m is accepted as the valid PHI.

Security Analysis

In this section, we analyze how the proposed scheme can effectively meets with the design goals presented in "System Model".

The proposed protocol achieves data security and access control. The essential characteristics of blockchain guarantees immunity of the proposed protocol. In other words, the data stored in the blockchain are unchangeable unless 51% attack[5] happens [12]. This ensures that the data can not be modified. Additionally, the PHI is encrypted under the data owner's public key. So it can only be decrypted by the data owner. Then, the encrypted PHI is stored in the private blockchain. Only authenticated accessors are allowed to obtain the ciphertext, which enhances security

[5] 51% attack brings the attacker more cost than benefits thus it rarely happens [12].

of the data. Moreover, data integrity can be achieved by the signature of the block generator in each block.

Furthermore, in order to efficiently utilize the data, the data owner may allow the authorized doctor to access the original PHI by sending a keyword trapdoor T_w to the doctor, as described in "Protocol description". Note that in $T_w = (T_1, T_2, T_3)$, $T_1 = Y_j/(y_{i1} + H_1(w) + y_{i3}H_6(\beta, r))$ is related with the public key Y_j of the intended doctor j. Only doctor with private key y_j is able to compute $U_1 = \hat{e}(A + H_6(\beta, T_3)E, T_1)^{1/y_j}$ and $U_2 = \hat{e}(B, T_2)^{1/(y_j H_1(w))}$, which are used to search for the keyword w and decrypt the original PHI. By this method, the data owner control the access of his/her PHI.

On the other hand, the hospital can also control its data access. As described in *Phase 3*, the doctor is required to login in the server of hospital for accessing the block ID_b in the private blockchain, which contains the requesting encrypted PHI. Consequently, the hospital is able to control the access of the data.

The proposed protocol achieves privacy preservation. Apart from data security, the proposed scheme achieves privacy preservation by anonymity. The encrypted PHI is appended with data owner's pseudo identity, which is generated by the data generator (doctor). Recall that the pseudo identity $d_i = (A_d, B_d, E_d, F_d)$, where $A_d = r_0(Y_{i1} + H_1(ID_i)P_1) + r_1P_1$ and $B_d = r_1Y_{i2}$. The random numbers r_0 and r_1 ensure that eavesdroppers can not deduce the owners of the PHI. Also, they can not link different flows of data with the same patients.

The proposed protocol achieves secure search. In *Phase 2* of the protocol, health record is encrypted with keyword search. In *Phase 3*, a doctor may get a searching trapdoor from a patient in order to searching for his/her history health record for diagnosis improvement. Recall that in searching trapdoor $T_w = (T_1, T_2, T_3)$, the term $T_1 = Y_j/(y_{i1} + H_1(w) + y_{i3}H_6(\beta, r))$ includes the public key Y_j of the authorized doctor j. Therefore, only doctor j with private key y_j is able to calculate U_1. Also, as T_w is a function of the keyword w, the doctor is only authorized to access the health record with intended keyword w without revealing user's other health information. Even though an eavesdropper eavesdrops the trapdoor, he cannot guess the keyword. This is because T_w involves the secret key of the patient and the doctor, which helps to resist the keyword guess attack.

The proposed protocol achieves time controlled revocation. As described in *Phase 3* of the protocol, an enable function $f(x)$ is introduced in the trapdoor to control the access right of the doctor within an allowable time window. The function (5) determines that only when $x_1 \geq x_2$ the value of $f(x_1)/f(x_2) = 1$, where x_1 and x_2 can be set as the timestamp. Specifically, in $U_1 = \hat{e}(A + H_6(\beta, T_3/f(t_0))E, T_1)^{1/y_j}$, the term $H_6(\beta, T_3/f(t_0)) = H_6(\beta, rf(t_s)/f(t_0)) = H_6(\beta, r)$ holds if the block generation time $t_0 < t_s$, which means that the data is generated before the trapdoor generation time. In other words, the doctor is only allowed to find out the user's records that are generated before the time t_s (current time).

The proposed protocol achieves system availability. In the consortium blockchain, the consensus mechanism requires that the transaction senders provide an evidence to show that the secure keywords are selected from the allowable set. The evidence $c_e = (A, J, \mathbf{X})$ concludes the keyword ciphertext A and the verification vector $\mathbf{X}$. By checking $\hat{e}(A, P_1) = \hat{e}(X_1, P_1) \cdot J$, it guarantees that the keyword contained in $\mathbf{X}$ and A is identical. The verification equation $\hat{e}(\mathbf{a}\cdot\mathbf{X}, X_2) = \hat{e}(X_1, X_1)$ means $\mathbf{a}\cdot\mathbf{X} = r_1P_1$, which ensures that the keyword is selected from the predefined set. Moreover, $\hat{e}(r_1P_1, X_2) = \hat{e}(X_1, X_1)$ makes sure that $X_2 = H_1(w)X_1$. Similarly, by checking whether $\hat{e}(X_i, X_{l-i}) = \hat{e}(X_k, X_{l-k})$, $1 \leq i, k, l \leq n$, the verifiers can find out whether $X_l = (H_1(w))^{l-i}X_i$ (Let $l > i$).

In the private blockchain, the pseudo identity for each patient is generated by the doctor from the real identity. Take the pseudo identity d_i of ID_i as an example, the identity ID_i is encrypted with ID_i itself as the keyword in order to ensure that the ciphertext d_i is searchable by the intended doctor. The user i can authorize his/her doctor to search for the secure indexes from his/her pseudo identities by sending an identity searching trapdoor to the doctor. In this way, the doctor can find out which secure keywords belong to the interested patients. Notably, in the identity searching trapdoor $T_d = (T_1^d, T_2^d, T_3^d)$, $T_1^d = Y_j/(y_{i1} + H_1(ID_i) + y_{i3}H_6(\beta, r_d))$ is related with the public key Y_j of the doctor. Therefore, only the authorized doctor j with T_d is allowed to find out the pseudo identities of the user.

Implementation and Performance Evaluation

In this section, we implement the proposed BSSP on JUICE[6], which is an open service platform serving for the study of designing smart contract and blockchain-based applications. This platform supports *Solidity* (designed for writing contracts) as Ethereum. It also provides friendly web/client management and monitoring tools based on *Java* and *Javascript*. Firstly, we illustrate the parameter settings and platform. Then, we compare the security properties among the proposed protocol and several other protocols. Later, the storage overhead and communication overhead are analyzed. Finally, the proposed *BSPP* is implemented on JUICE platform to evaluate its performance.

[6] https://www.juzhen.io/

Parameter Settings and Platform

The system parameter $\lambda = 128$. We use Type A pairing on the elliptic curve $y^2 = x^3 + x$ over the field $\mathbb{F}_p$ for some prime $p = 3 \ mod \ 4$. The cryptographic primitives are implemented on a computer with Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ, 3 GB RAM, Microsoft Windows 7 operating system, using Java language, as shown in Table 4. JPBC library [7] is used for the simulation.

We utilize JUICE (*client version*) to build a private test chain comprising only permission nodes in three servers. The configurations are shown in Table 5. In the main node, *nginx-1.11.3* (a high performance HTTP and reverse proxy server), *truffle-2.1.1* (a development framework for the Ethereum) and *JUICE-client* (a client version of JUICE) are deployed. The other two nodes only need to deploy *truffle-2.1.1* and *JUICE-client*. The *JUICE-client* Software (Microsoft Windows Version) is installed for managing and monitoring transactions in the chain. We skip the details of the deployment process due to space limitations.

Note that transactions are published in the local computer (see Table 4 for the configuration) via a permission node (No.1 is chosen in our simulation). The transaction can be easily put on the test chain without writing additional integration codes for Java and Solidity because of *Web3j* (a lightweight library for Java applications). Since Solidity only provides *now*[8] accuracy of one second, the time cost is obtained by using *shell* script and *javascript* instead. All simulations are implemented 500 times for average.

Comparisons of Security Properties

Due to the fact there is no blockchain-based PHI sharing scheme for diagnosis improvements up to this end, we choose recently proposed medical record sharing protocols [3, 11, 13, 17, 18] as benchmarks.

Table 6 compares the security properties of the proposed scheme with blockchain based schemes Xia-I [11], Xia-II [13], Peterson [18], and non-blockchain based schemes Yang [3], Zhang [17]. From the table we can find that only the proposed scheme and Yang [3] achieve both searchability and time controlled revocation. Notably, all the schemes have the properties of access control, data auditing and privacy preservation, which are the critical security objectives in health record sharing systems.

[7] http://gas.dia.unisa.it/projects/jpbc/#.Wm8S_GWHnKo

[8] http://solidity.readthedocs.io/en/develop/units-and-global-variables.html

Table 4 Simulation platform

Operating system	Ubuntu 16.04
CPU	Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ
Memory	3 GB RAM
Program language	Java

Storage Overhead and Communication Overhead Analysis

Storage overhead. The computer clients in a hospital store the blocks for the private blockchain of the hospital. The hospital servers store the blocks for the consortium blockchain. The overall storage overhead is in proportion to the number of the transactions, which is dynamic increasing with the time. In this subsection, we analyze the size of each kind of block, which is the storage overhead of each transaction.

We denote $|G_1|$ and $|G_2|$ the size of an element in group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, $|Q|$ the size of an element in $\mathbb{Z}_p$. The size of β, ID_j, $j \in \mathcal{D}$, signature, timestamp is 9 bytes, respectively. The size of block size is 4 Bytes. The size of previous block hash value, encrypted PHI hash value, block identity is 32 Bytes, respectively. As shown in Table 2, the user pseudo identity $d_i = (A_d, B_d, E_d, F_d)$, where A_d, B_d, E_d are elements in group $\mathbb{G}_1$. They have the size of $3|G_1|$. F_d is an element in $\mathbb{Z}_p$. They totally have the size of $3|G_1| + |Q|$. In the secure keyword $(c_{i1}, c_{i2},)$. the term c_{i1} has the same size with user pseudo identity, i.e.,$(3|G_1| + |Q|)$. For $c_{i2} = (J, \mathbf{X})$, J is an element in $\mathbb{G}_2$ and $\mathbf{X}$ is a vector with n elements in $\mathbb{G}_1$. Thus, the size of c_{i2} is $n|G_1| + |G_2|$. Based on the above analysis, the size of each block in the private blockchain is $(n+6)|G_1 + |G_2| + 2|Q| + 127$.

Regarding the blocks in consortium blockchain, the block generator ID ID_k is 9 Bytes. In the secure indexes, each index $Tx_i = (ID_b, d_i, c_{i1})$ has the size of $6|G_1|+2|Q|+9$. Therefore, the storage overhead of each block in consortium blockchain is $8(6|G_1| + 2|Q| + 9) + 95$. Table 7 illustrates the storage overhead of per block in computer client and the server.

Communication overhead. According to Fig. 4, the communication overhead comes from two stages: Data generation stage, data search and access stage. At data generation stage, the overall communication overhead is in proportion to the number of blocks generated by the doctors. This number is also dynamic, thus we analyze the communication overhead for each block generation step. Firstly, the computer client broadcasts a new transaction TX with η, where TX is composed by data generator ID ID_j, user pseudo identity d_i, secure key words (c_{i1}, c_{i2}), encrypted PHI hash, and contributor signature σ_j. It brings

Table 5 Configurations of nodes in the test chain

No.	Operating system	CPU	Memory	LAN IP	Server configuration
1	Ubuntu 16.04	Intel(R) Xeon(R) CPU E5-2603 v4 @ 1.70 GHz	8 GB RAM	192.168.1.237	*nginx-1.11.3*;*truffle-2.1.1*; *JUICE-client*
2	Ubuntu 16.04	Intel(R) Xeon(R) CPU X3320 @ 2.50 GHz	4 GB RAM	192.168.1.238	*truffle-2.1.1*;*JUICE-client*
3	Ubuntu 16.04	Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60 GHz	32 GB RAM	192.168.1.239	*truffle-2.1.1*;*JUICE-client*

Table 6 Comparisons of security properties

Properties	Yang [3]	Zhang [17]	Xia-I [11]	Xia-II [13]	Peterson [18]	Proposed BSPP
Blockchain-based	×	×	√	√	√	√
Access control	√	√	√	√	√	√
Data auditing	√	√	√	√	×	√
Privacy preservation	√	√	√	√	√	√
Secure search	√	×	×	×	×	√
Time controlled revocation	√	√	×	×	×	√

Table 7 Storage overhead of per block

Entity	Computer client	Server
Storage overhead	$(n+6)\|G_1\|+\|G_2\|+2\|Q\|+127$	$48\|G_1\|+16\|Q\|+167$

Table 8 Communication overhead for per block

Phases	Data generation		Data search and access	
	Data broadcast	Data verification	Data search	Data access
Private blockchain	$(n+6)\|G_1\|+\|G_2\|+3\|Q\|+59$	$13\times\lfloor\frac{2}{3}n_p\rfloor$	-	32
Consortium blockchain	$8((n+6)\|G_1\|+\|G_2\|+2\|Q\|)+90$	$13\times\lfloor\frac{2}{3}n_c\rfloor$	$6\|G_1\|+2\|Q\|$	-

Table 9 Time cost (in *ms*) of cryptographic algorithms

$n=500$								
Algorithms	*buildSystem*	*encrypt*	*verPrivate*	*verConsortium*	*trapdoorGen*	*searchID*	*searchW*	*decrypt*
Max Time	135	4870	25	4892	43	46	24	6
Min Time	47	4578	16	1000	34	19	19	0
Average Time	50	4620	18	4565	36	20	20	0
				$n=1000$				
Max Time	134	10751	23	10218	45	32	27	2
Min Time	47	8990	16	100	34	19	19	0
Average Time	51	9278	18	9241	36	21	21	0

Table 10 Time cost (in ms) for publishing transactions

Blockchains	Private blockchain	Consortium blockchain	
		$n = 500$	$n = 1000$
Tx length (in Bytes)	3674	32882	64882
Max time	5954	6950	8484
Min time	1134	2315	3277
Average time	5584	6569	8164

$(n + 6)|G_1| + |G_2| + 2|Q| + 50$ communication cost. The evidence $\eta = (\alpha, \beta')$ yields $|Q| + 9$ communication overhead. Thus the total communication overhead is $(n + 6)|G_1| + |G_2| + 3|Q| + 59$. The transaction is accepted by the blockchain after getting $\lfloor \frac{2}{3} n_p \rfloor$ validation confirm message. Each validation confirm message contains the block identity (9 Bytes) and an indication of validation (4 Bytes). Thus, the total communication overhead caused by the verification process is $13 \times \lfloor \frac{2}{3} n_p \rfloor$. In the consortium blockchain, publishing a transaction $TX = (ID_k, Tx_1, Tx_2, \cdots, Tx_8, \sigma_k)$ with the 8 ciphertext c_{i2} yields $8((n+6)|G_1|+|G_2|+2|Q|+9)+18$ communication overhead. Similarly, the verification process causes $13 \times \lfloor \frac{2}{3} n_c \rfloor$ overhead, where n_p denotes the amount of the nodes in the consortium blockchain.

In the data search stage, the patient sends an identity searching trapdoor T_w^d and a keyword searching trapdoor T_w to the hospital. This transmission generates $6|G_1| + 2|Q|$ communication overhead. After finding out the interested block identity ID_b, the doctor will access the corresponding private blockchain and ask for getting the interested ciphertext c_{i0}. The communication overhead of this process is 32. The communication overhead of different phases is shown in Table 8.

Given the system parameter $\lambda = 128$, the lengths of $\mathbb{Z}_p$, $\mathbb{G}_1$ and $\mathbb{G}_2$ are 256 bits, 512 bits, and 3072 bits, respectively. Under these settings, the storage overhead of the computer client and the server is $2|Q| + (n + 6)|G_1| + |G_2| + 127 = (64 * n + 959)$ Bytes and $48|G_1| + 16|Q| + 167 = 3751$ Bytes respectively. The communication overhead for data broadcasting in the private blockchain and consortium blockchain is $(n+6)|G_1|+|G_2|+3|Q|+59 = (64n+923)$ Bytes and $8((n+6)|G_1|+|G_2|+2|Q|)+90 = (512n+6746)$ Bytes, respectively. From the results we can find out that the storage overhead of the computer client is in proportion to the size of the keyword set. Also, the communication overheads of data broadcasting in both private blockchain and consortium blockchain grow with n.

Implementation and Time Cost Evaluation

As the JUICE has not yet provided the API of public encryption with keyword search, we measure the time cost of cryptographic primitives and transaction publishing separately.

Implement of cryptographic primitives. The cryptographic algorithms in the three phases are implemented on the platform shown in Table 4. We record the time cost of the algorithms in Table 9. The algorithms *GlobalSetup* and *KeyGen* in system setup phase are finished by the algorithm *buildSystem* in our implementations. In order to generate a valid transaction, a computer client is required to compute the ciphertext c, user pseudo identity d_i, a proof η[9]. The algorithm *encrypt* is responsible for this task. As the time cost for calculating vector X varies with the size of keyword set n. We implement the algorithms by setting $n = 500$ and $n = 1000$. The verification algorithms *verPrivate* and *verConsortium* check the validity of the new coming data in the private blockchain and consortium blockchain. The trapdoors T_d and T_w are generated by algorithm *trapdoorGen*. The algorithms *searchID* and *searchW* search for the intended user's identity and and the intended keyword. The ciphertext c_{i0} is decrypted by *decrypt*.

From Table 9, we can find out that the average time of generating a validate transaction at the computer client increases with the size of the keyword set n. It is caused by the vector $X = [X_1, X_2, \cdots, X_n]$, which requires n times scalar multiplication in $\mathbb{G}_1$. Correspondingly, the verifiers in the consortium blockchain also have to perform n times scalar multiplication in $\mathbb{G}_1$, which brings time overhead growing with n. The time cost of the other algorithms are not effected by n. Notably, the table shows that the average time for decrypting the ciphertext is 0. This is because this algorithm only requires one hash, one XOR and one exponentiation operations in $\mathbb{G}_2$. The computational time is far lower than $1ms$. Thus, the system outputs 0 for this algorithm.

Publishing transactions in the blockchain. The time cost of publishing a transaction is effected by the length of the package. Therefore, we firstly calculate the size of the

[9]The computer also needs to compute a signature. As signature algorithm is not specified in the scheme, we do not consider its time cost in the system.

transactions in the blockchains. As analyzed in the above subsection, the lengths of a transaction in private blockchain and consortium blockchain are $l_p = (n+6)|G_1| + |G_2| + 2|Q| + 50$ and $l_c = 8(6|G_1| + 2|Q| + 9) + 18$, respectively. Substituting $|G_1|, |G_2|, |Q|$ with $64, 384, 32$, we obtain $l_p = (64n + 882)$ Bytes and $l_c = 3674$ Bytes. The transactions are published by padding the data with length l_p and l_c in the defined transaction format of JUICE. As l_p varies with n, we implement the simulations by setting $n = 500$ and $n = 1000$ in the consortium blockchain, respectively. The results are illustrated in Table 10.

In the table, when the TX length jumps from 3674 Bytes to 32882 Bytes, the average time cost does not change that much, from $5584ms$ to $6569ms$. But when the TX length is 64882 Bytes, the time cost arrives $8164ms$. The reasons can be found out by analyzing the processes of transaction publishing. It includes padding the transaction into a package, signing the package, and broadcasting it. After being verified by other nodes, the TX is accepted and added to the chain. For each transaction, these steps yield a basic time cost, which is about $5000ms$ in our simulations. The time cost increases slowly with the growth of the package length because the time cost of signature, transmission, and verification processes increase with the package length. This result demonstrates that the size of the keyword set should not be too much large in oder to control the efficiency of the transaction. But it should be not too small because of system availability.

Conclusions

We have proposed a blockchain based secure and privacy-preserving PHI sharing protocol (BSPP) for diagnosis improvements in e-Health system. Firstly, two blockchains, i.e., private blockchain and consortium blockchain, are introduced and designed in the system to realize health record sharing. Furthermore, proof of conformance is defined and devised for the blockchains as the consensus mechanism to construct validated blocks. Based on the blockchains, the PHI sharing protocol is proposed by using public key encryption with keyword search. After getting trapdoors from the patient, the doctor is authorized to search and access the intended history health records for improving the diagnosis. Security analysis demonstrates that the proposed protocol achieves data security, privacy preservation, secure search and time controlled revocation. Furthermore, we implement the scheme on JUICE and evaluate the performance from the aspects of storage overhead, communication overhead, and time overhead.

For future work, we will develop a specific miner and verifier election algorithm for the e-Health blockchains. Also, we will consider conjunctive keyword search.

Acknowledgments This work is partly supported by the National Natural Science Foundation of China (Grants No. 61601005, No. 61571240), Natural Science Foundation of Anhui Province (Grant No. 1608085QF138, No. 1808085MF164), Anhui Provincial Key Laboratory of Network and Information Security (Grant No. AHNIS2018003), and Scientific Research Staring Foundation of Anhui Normal University (Grant No. 2014bsqdjj38).

Compliance with Ethical Standards

Conflict of interests Author Aiqing Zhang declares that she has no conflict of interest. Author Xiaodong Lin declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abbas, A., and Khan, S. U., A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. *IEEE Journal of Biomedical and Health Informatics* 18(4):1431–1441, 2014.
2. Shen, Q., Liang, X., Shen, X., Lin, X., and Luo, H., Exploiting geo-distributed clouds for a e-Health monitoring system with minimum service delay and privacy preservation. *IEEE Journal of Biomedical and Health Informatics* 18(2):430–439, 2014.
3. Yang, Y., and Ma, M., Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-Health clouds. *IEEE Transactions on Information Forensics and Security* 11(4):746–759, 2016.
4. Zhou, J., Cao, Z., Dong, X., and Lin, X., PPDM: A Privacy-preserving protocol for cloud-assisted e-Healthcare systems. *IEEE Journal of Selected Topics in Signal Processing* 9(7):1332–1344, 2015.
5. Zhang, Z., Dong, M., Zhu, L., Guan, Z., Chen, R., Xu, R., and Ota, K., Achieving privacy-friendly storage and secure Statistics for smart meter data on outsourced clouds, *IEEE Transactions on Cloud Computing*. https://doi.org/10.1109/TCC.2017.2685583.
6. Chang, S., Zhu, H., Dong, M., Ota, K., Liu, X., and Shen, X., Private and flexible urban message delivery. *IEEE Transactions on Vehicular Technology* 65(7):4900–4910, 2016.
7. Esposito, C., De Santis, A., Tortora, G., Chang, H., and Choo, K. K. R., Blockchain: a panacea for healthcare cloud-based data security and privacy?. *IEEE Cloud Computing* 5(1):31–37, 2018.
8. Novo, O., Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal* 5(2):1184–1195, 2018.
9. Wang, J., Li, M., He, Y., Li, H., Xiao, K., and Wang, C., A blockchain based privacy-preserving incentive mechanism in crowdsensing applications. *IEEE Access* 6:17545–17556, 2018.
10. Dorri, A., Steger, M., Kanhere, S. S., and Jurdak, R., Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine* 55(12):119–125, 2017.
11. Xia, Q., Sifah, E., Smahi, A., Amofa, S., and Zhang, X., BBDS: Blockchain-Based data sharing for electronic medical records in cloud environments. *Information* 8(44):1–16, 2017.
12. Kuo, T., Kim, H., and Ohno-Machado, L., Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association* 24(6):1211–1220, 2017.
13. Xia, Q., Sifah, E. B., Asamoah, K. O., Gao, J., and Du, X., MeDShare: Trust-less medical data sharing among cloud service

providers via blockchain, *IEEE Access*. https://doi.org/10.1109/ACCESS.2017.2730843.
14. Yue, X., Wang, H., Jin, D., Li, M., and Jiang, W., Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *Journal of Medical Systems* 40(10):218, 2016.
15. Zyskind, G., Nathan, O., and Pentland, A., Decentralizing privacy: Using blockchain to protect personal data. *IEEE Security and Privacy Workshops*: San Jose, 18–20, 2015.
16. Azaria, A., Ekblaw, A., Vieiraand, T., and Lippmanl, A., Medrec: Using blockchain for medical data access and permission management. *IEEE International Conference on Open and Big Data*, 25–30, 2016.
17. Zhang, J., Xue, N., and Huang, X., A secure system for pervasive social network-based healthcare. *IEEE Access* 4(99):9239–9250, 2016.
18. Peterson, K., Deeduvanu, R., Kanjamala, P., and Boles, K., A blockchain-based approach to health information exchange networks.
19. Shae, Z., and Tsai, J., On the design of a blockchain platform for clinical trial and precision medicine. *International Conference on Distributed Computing Systems (ICDCS 2017)*: Atlanta, 2017.
20. Zhao, H., Zhang, Y., Peng, Y., and Xu, R., Lightweight backup and efficient recovery scheme for health blockchain keys. *IEEE International Symposium on Autonomous Decentralized System (ISADS)*: Bangkok, 22–24, 2017.
21. Boneh, D., Crescenzo, G. D., Ostrovsky, R., and Persiano, G., *Public key encryption with keyword search, EUROCRYPT 2004, LNCS*. Vol. 3027, pp. 506–522. Berlin: Springer, 2004.
22. Baek, J., Safavi-Naini, R., and Susilo, W., Public key encryption with keyword search revisited, *International Conference on Computational Sciences and its Applications (ICCSA)*: Perugia, 2008.
23. Hu, C., and Liu, P., An enhanced searchable public key encryption scheme with a designated tester and its extensions. *Journal of Computer* 7(3):716–723, 2012.
24. Shao, J., Cao, Z., Liang, X., and Lin, H., Proxy re-encryption with keyword search. *Information Science* 180(13):2576–2587, 2010.
25. Yau, W., Phan, R., Heng, S., and Goi, B., Proxy re-encryption with keyword search: New definitions and algorithms. *International Conference, SecTech and DRBC*: Jeju Island, 13–15, 2010.
26. Ogata, W., and Kurosawa, K., Oblivious keyword search. *Journal of Complexity* 20(2-3):356–371, 2004.
27. Ryu, E., and Takagi, T., Efficient conjunctive keyword-searchable encryption. *IEEE 21st International Conference on Advanced Information Networking and Applications*: Niagara Falls, 21–23, 2007.
28. Bethencourt, J., Song, D., and Waters, B., New constructions and practical applications for private stream searching (extended abstract). *IEEE Symposium on Security & Privacy*: Berkeley, 21–24, 2006.
29. Boneh, D., and Waters, B., *Conjunctive, subset, and range queries on encrypted data, TCC 2007, LNCS*. Vol. 4392, pp. 535–554. Berlin: Springer, 2007.
30. Wang, X., Huang, X., Yang, X., Liu, L., and Wu, X., Further observation on proxy re-encryption with keyword search. *The Journal of Systems and Software* 85:643–654, 2012.
31. Castro, M., and liskov, B., Practical Byzantine Fault Tolerance, *the Third Symposium on Operating Systems Design and Implementation*: New Orleans, 1999.
32. HL7. HL7 Fast Healthcare Interoperability Resources (FHIR). https://www.hl7.org/fhir/. Accessed: 2017-11-20.
33. Leftwich, R., The path to deriving clinical value from FHIR - InterSystems, http://www.intersystems.com/library/library-item/path-deriving-clinical-value-fhir/. Accessed: 2017-11-20.