A Blockchain-based Approach to the Secure Sharing of Healthcare Data

Huihui Yang, Bian Yang
Department of Information Security and Communication Technology, CCIS, Norwegian University of Science and Technology, Norway

Abstract
Electronic health record (EHR) sharing enables to improve the quality and reduce the cost of healthcare, but it is still challenging because of technique issues even though patients and healthcare organizations are willing to share. These technical barriers include confidentiality, privacy, interoperability, integrity and so on. In this paper, we propose a blockchain and MedRec-based approach by implementing signcryption and attribute-based authentication to enable the secure sharing of healthcare data. By implementing this approach: 1) all patient’s fragmented EHR pieces can be viewed as a whole record and stored secure against tampering; 2) the authenticity of patients’ EHRs can be verified; 3) flexible and fine-grained access control can be provided and 4) maintaining a clear audit trail is possible.

1 Introduction
During medical care service, large amount of data are created and need to be stored safely for a long period, often a life time. One major difference between healthcare data and other big data sharing is that electronic health records (EHRs) are normally highly sensitive, which may make patients and medical organizations reluctant to share. On the other hand, however, EHR sharing [17, 19] can benefit both patients and medical organizations in a few ways. First of all, data sharing can facilitate medical research, for example, pooling multiple medical trials together for better understandings and scientific discoveries. Secondly, collaboration between different healthcare organizations (even cross-boarder cases) will be easier, such as doctors accessing patient’s medical records, the reimbursement of medical treatment in a foreign country and so on. Thirdly, regulations and standards to facilitate secure EHR sharing will be developed and strengthened, which will in turn bring more trust among different medical organizations and thus can offer patients better service and further improve medical research as well. However, there still exist some technical barriers that hinder EHR sharing and make it challenging in many ways. According to a literature survey [13] on research papers about main security issues related to EHRs sharing from the year 2004 to 2014, the main eight security issues (from the

This paper was presented at the NIK-2017 conference; see http://www.nik.no/.
most concerned to the least) are confidentiality, privacy, access control, integrity, data authenticity, user authentication, audibility and transparency. It is urgent to solve these issues to enable secure EHR sharing.

Related work

During recent years, blockchain [5, 21] has been brought in the area of medical healthcare to solve security issues such as integrity and data authenticity. A blockchain is a distributed ledger to record transactions between parties. One of its most promising features is its decentralized structure. Storing EHRs on a centralized server may attract attacks from malicious parties, and the failure of an important site will probably cause service failure or interruption. However, a decentralized blockchain-based system has high probability to avoid this issue and thus makes it easier and more possible to provide continuous service. Another advantage of blockchain is that, due to the security of hash functions [14] and the proof-of-work (PoW) [15], it is hard to tamper transactions stored in the blocks.

As far as we know, there are not so many research results in blockchain-based EHR sharing. In [12], an approach for health information exchange network is proposed. There are two main contributions. First of all, it suggests solving the issue of interoperability by EHR semantic and format checking to ensure that all EHRs in the blockchain network have the same format. Secondly, it has proposed an algorithm to randomly choose the next miner to save computation power and system resources. In addition, [12] suggests using blockchain encryption, privacy preserving keyword searches and smart contracts to provide privacy and anonymity, but no detailed approaches included. The medical system MedRec proposed in [3] is built on contracts for easier management of EHRs. Users’ identification strings are mapped to their Ethereum addresses in registrar contracts (RCs) to keep users anonymous. Patient-provider relationship contracts (PPRs) define how data are managed and accessed; PPRs are referenced by links contained in summary contracts (SCs) so that all medical records of a patient can be united as a whole piece. However, [3] does not provide any detailed approaches to solve issues such as how patients’ EHRs are encrypted and accessed by authorized users, how users are authenticated and how to maintain an audit log about EHRs’ accessing. In [20], a blockchain based App HGD (Healthcare Data Gateway) is developed to provide patients an easy way to control and manage their medical data. All data are stored in the blockchain cloud and managed through the data management layer. Only the authorized can access patients’ data, the replica of which may be enforced to be destroyed when the authorized period is over. [7] focuses on discussing about the main aspects of medical records, and the advantages and disadvantages of using blockchain on the storage and retrieval of medical records.

Goals

It is very challenging to solve all the security and privacy issues (confidentiality, privacy, access control, integrity, data authenticity, user authentication, auditability, transparency and interoperability) mentioned earlier in this section. As far as we know, there is no literature work that studies on detailed approaches how to solve these issues by technique means such as encryption and authentication. This is also our main goal in this paper. We first adopt the smart contract based model MedRec proposed in [3], based on which we use signcryption and attribute-based authentication (ABA) to
provide: (1) Data authenticity: the authenticity of patients’ EHRs can be verified by those who access the data. (2) Data integrity: patients’ EHRs can be stored secure against tampering. (3) Data confidentiality: patients’ EHRs are stored secure and kept secret from the unauthorized. (4) Flexible access control: patients can decide how their EHRs are accessed and only those who are authorized can access patients’ EHRs. (5) Authentication: users should be authenticated before they access EHRs. (6) Audit trail: how patients’ EHRs are accessed is kept in an audit trail that patients can access.

This paper is organized as follows. In Section 2, we will briefly describe the main entities in the system and give a big view of the MedRec-based system structure. In Section 3, details of our approach will be described, such as some background knowledge that will be used in our approach, EHR sharing and updating, user authentication and incentive measures. Then in Section 4, we discuss security issues related to the proposed approach. We conclude this paper and the main innovations in the last section.

2 A Big Picture of the MedRec-based Approach

To better understand the proposed blockchain-based approach for EHR secure sharing, a big picture will be presented in this section, including main entities and the main structure.

Main entities

Before introducing the system, we first describe main entities and their roles so as to have a big view of the system construction.

- **Patient** A patient is a person who seeks for healthcare services from medical, health and welfare organizations. Patients’ privacy and the security of their EHRs are the main concerns in this paper.

- **Service provider** There are two types of service providers, i.e., those that provide health and welfare services (denoted by $SP_1$) and those that only provide storage services (denoted by $SP_{II}$).

- **Users** Patients are only one type of users. To avoid mixing patients with those who access patients’ EHRs, when we say users, it refers to the latter. Thus, a patient can also become a user when he accesses another patient’s EHRs.

- **Attribute issuer** An attribute issuer is an entity that issues attribute keys to users and patients. Instead of one trusted attribute issuer, multiple and parallel attribute issuers are used. The main reason is that we want to avoid any centralized entity that may become the bottleneck or attack target of malicious parties.

The main structure

One main goal of our approach is to provide patients with better and easier management for their fragmented EHR pieces, which can be implemented based on MedRec proposed in [3]. As shown in Fig. 1(a), the patient is using services from service providers $A$ and $B$ (denoted by $SP_A$ and $SP_B$ respectively), and the data generated by each service provider are stored in their own databases denoted by $DB_A$ and $DB_B$. Considering the sizes of EHRs, only contracts (such as PPRs) but not data will be stored on the blockchain, where contracts are data structures by following which a patient can access his data stored on service providers’ database.
The process how a patient store and access his data is as follows. When a patient (i.e., patient A) first joins the blockchain-based healthcare network, he first registers himself through RC. To provide services, providers A and B also need to register themselves in RC. From the entries in RC, patients can link to their own SCs, the summaries with references to PPRs, which define the way how a provider manages patients’ EHRs and how EHRs can be accessed. Since PPRs are an assortment of data pointers that consist of query strings, the patient can execute the query strings and then get related records. Therefore, when patient A uses the service provided by $SP_A$, the generated data can be stored in either $DB_A$ or on the cloud. For the latter case, we still consider it as a special service (i.e., cloud storage), so we will only discuss the first situation here. To store data in $DB_A$ and related contracts on the blockchain, $SP_A$ first generates related PPRs and send them as part of the transactions in the next block that will be added in the blockchain. Once a new block is mined and verified, patient A’s PPRs will be added as transactions in the newly mined block, and the links to the new PPRs will also be added in SCs. Meanwhile, the service provider will enforce EHR changes in its database. In this way, all EHR changes will be updated and recorded in the blockchain. For instance, data deletion, data stored in the database will be deleted, but the action of deleting and the agreement to deletion from the patient’s side will also be considered as contracts, which will also be uploaded to the blockchain. Data pieces stored separately can be referenced by different PPR entries with links in SCs. As a result, even though patient A is using services from different service providers and EHRs stored separately, all these fragmented data pieces can still be connected and be viewed a whole piece (Refer to [3] for more details). To keep patients’ data secure, all blocks uploaded to the blockchain should be encrypted and later can be searched by privacy preserving keyword searches [10, 11, 16].

3 Signcryption-based EHR Sharing

Details about how to use cryptographic primitives to achieve security and privacy in our approach will be presented in this section. Since the EHR secure sharing is based on signcryption and ABA, we will first briefly introduce them in Subsection 3, then the signcryption-based data sharing and accessing in Subsection 3 and EHR updating in Subsection 3. At last, user authentication and blockchain related incentive measures will be discussed in Subsections 3 and 3 respectively.
Prerequisites

Signcryption [9] is a combination of encryption and signature, i.e., data is first encrypted and then signed or the opposite sequence, so as to provide both confidentiality and data authenticity. In this paper, we choose the combination of attribute-based encryption (ABE) [2] and PKI-based signature, with the sequence of first encryption and then signature.

ABA [6, 8, 18] is an authentication approach based on attributes. To get authenticated, a signer generates a signature using attributes required in an attribute predicate, and only those who possess the required attributes can generate a valid signature. Given a valid signature, what the verifier knows is only that the signer owns the required attributes, and therefore anonymous authentication can be achieved. If the signer’s identity is required in special cases (e.g., being used as a legal evidence), the signature can be opened by an authority and the signer’s identity information can be revealed.

EHR sharing and accessing

As described in Subsection 2, EHRs are not stored on blockchain but Internet nodes owned by service providers or cloud servers. If these service providers are not fully trusted, EHRs should be encrypted before being uploaded to the servers. In addition, to ensure data authenticity, the encrypted data should also be signed by the owner. The private key the owner uses to sign is generated based on its Ethereum address. Since a user can have as many addresses as he wants to, the signature will not reveal his identity or hinder his privacy. To achieve both privacy and data authenticity, signcryption [1] will be applied in our approach.

The main idea of Signcryption used in our approach can be illustrated in Fig. 2. Suppose there is a patient who wants to share his EHRs with other users (e.g., doctors, nurses and his family members). In order to control those who can access his EHRs, the patient needs to define the access policies, which are presented by attribute predicate $\Psi$. For simplicity, $\Psi$ is presented by attribute keys denoted by $\text{AttKey}_i$ ($1 \leq i \leq m$) that are issued by attribute issuer $\text{AI}_i$. The public and private key pair of the patient is $pk$.
and \( sk \). One thing to notice is that this public and private key pair is generated based on one of his addresses, and thus will not reveal his identity information. Assume the attribute issuers trusted by the patient is denoted by \( \{ AI_1, \ldots , AI_m \} \), and the patient will encrypt his EHRs by attribute keys issued by these attribute issuers. Then the EHR sharing and access proceeds as follows:

1) **Attribute key issuing**: users communicate with attribute issuers to get private attribute keys, which correspond to public attribute keys that are published by the issuers. A user may obtain attribute keys from a single or multiple issuers. Assume a user \( U_k \) possesses attribute set \( \Phi_k = \{ att_{k1}, \ldots , att_{kn} \} \), where \( kn = |\Phi_k| \) is the size of \( \Phi_k \). The public attribute key of \( att_{kj} \) \( (1 \leq j \leq kn) \) obtained from the same issuer \( AI_i \) is the same, denoted by \( apk_{i,j} \), while users’ private attribute keys are different, denoted by \( ask_{i,k,j} \) for user \( U_k \).

2) **Public parameter obtaining**: before uploading encrypted EHRs to a server, a patient first needs to retrieve public parameters such as public attribute keys from attribute issuers (one or multiple) that he trusts.

3) **Data encryption, signing and uploading**: there are three main parts in this step, i.e., EHRs encryption, data key encryption and ciphertext signing. First of all, encrypt EHRs with data key \( K \) by a symmetric encryption algorithm \( E_1 \) (e.g., AES), and the ciphertext is denoted by \( C_1 = E_1(K,EHRs) \). Secondly, encrypt the data key \( K \) by ABE (denoted by \( E_2 \)) with attribute key set \( AttKey_i \), which denotes the required attribute keys issued by \( AI_i \) corresponding to attribute set \( \Phi \). Then the ciphertext of data key \( K \) encrypted by attribute key set \( AttKey_i \) is denoted by \( C_2 = E_2(AttKey_i,K) \). If the patient wants to share with users whose attribute keys are issued by multiple attribute issues (denoted by \( AI_1, \ldots , AI_m \)), he has to encrypt \( K \) with attribute key sets \( \{ AttKey_{1y}, \ldots , AttKey_{my} \} \) and thus the encrypted data key \( K \) is \( C_2 = \{ E_2(AttKey_{1y},K), \ldots , E_2(AttKey_{my},K) \} \). Thirdly, hash the concatenation of \( C_1 \) and \( C_2 \) and then sign the message digest with \( sk \), and the signature is \( \delta = Sign(sk, H(C_1,C_2)) \), where \( H \) is a secure one-way hash function. Finally, upload \{ \( C_1 \), \( C_2 \), \( \delta \) \} to the storage server.

One thing to notice is that the signcryption we use here is a simple combination of encryption and signature. However, one needs to carefully choose the ABE and the signature schemes to achieve better security and efficiency. An detailed example of cloud-based signcryption can be found in [9].

4) **Data accessing**: this step includes three parts: i.e., signature verification, data key and EHRs decryption. After downloading \{ \( C_1 \), \( C_2 \), \( \delta \) \}, the user first validates \( \delta \) to check data authenticity. If \( \delta \) is valid and the user possesses the attributes required in predicate \( \Psi \), he decrypts \( C_2 \) to get the data key \( K \). Finally, decrypt \( C_1 \) using \( K \) to recover the plaintext EHRs.

Due to the computation complexity of pairings, both the encryption and decryption process are more time consuming than symmetric algorithms. Therefore, large amount data such as EHRs are encrypted by symmetric algorithms and only the data key is encrypted by ABE. As described in Step 3), data key \( K \) is encrypted by different attribute key sets \( AttKey_i \) \( (1 \leq i \leq m) \), which infers that users with required attribute keys issued by \( AI_i \) can decrypt the encrypted EHRs. For
example, there are four attribute issuers \((A_{I1}, \cdots, A_{I4})\) and \(C_2 = \{E_2(K, \text{AttKey}_1), E_2(K, \text{AttKey}_2), E_2(K, \text{AttKey}_4)\}\), and a user \(u_k\) \((1 \leq k \leq 4)\) denotes one who is issued with the required attribute keys by \(A_{I1}\). Then only \(u_1, u_2\) and \(u_4\) can decrypt the encrypted \(EHRs\). Even if \(u_3\) has the required attributes, he still cannot decrypt \(C_1\) since he cannot decrypt any of \(C_{21}, C_{22}\) and \(C_{24}\) to get \(K\) with his attribute keys from \(A_{I3}\). Another possible case is that a user might possess the required attributes from multiple attribute issuers, and then he can use any of them to decrypt \(C_2\).

**EHR updating**

Any changes of patients’ \(EHRs\) should be updated and synchronized to the blockchain, such as \(EHR\) adding, deleting and modification, access policy changes, or the changes caused by attribute key updating and so on. In the following, we discuss these main updating cases separately.

- **EHR adding** New generated \(EHRs\) should be added and synchronized to the blockchain. The procedure of \(EHR\) adding is as illustrated in Fig. 3(a), including six steps: (1) the patient encrypts \(EHRs\), encrypts the data and then signs the ciphertext; (2) the patient uploads the signed ciphertext to the server; (3) the server sends the PPRs generated for the new \(EHRs\) to the blockchain as part of the transactions of the next block; (4) once a new block is mined and verified, the server changes the links to the new PPRs in SCs; (5) the server adds the signed ciphertext of \(EHRs\) in
its database; (6) the server sends a notification to the patient whether his EHRs are successfully added.

- **EHR deleting** To delete old EHRs stored on a server, the following needs to be done: (1) the owner (patient) of the EHRs needs to send a delete request to the server; (2) the server sends the EHR deleting information as part of the transactions of the next block; (3) once a new block is mined and verified, the server changes SCs to unlink the PPRs which reference to the EHRs to be deleted; (4) the server deletes patient’s EHRs; (5) the server sends a notification to the patient whether his EHRs are successfully deleted.

- **EHR modification** includes the following steps: (1) the patient encrypts the new data with the same data key and then signs it; (2) the patient uploads the signed ciphertext to the sever; (3) the server sends the modification and new PPRs to the blockchain; (4) once the changes are updated in the blockchain, the server unlinks the old PPRs and links the new PPRs if necessary; (5) the server deletes the old EHRs and adds the new; (6) the service sends a notification to the patients whether his EHRs are successfully modified.

- **Access policy change** refers to the situation that the data owner (the patient) has changed the rules how his EHRs can be accessed. Since the access policies are represented by attribute predicates, based on which the data key is encrypted. Therefore, the main changes that happen on the patient side is to re-encrypt the data key, while the EHRs does not need to be re-encrypted. The procedure goes as follows: (1) the patient downloads the encrypted EHRs first; (2) the patient encrypts the data key and signs the encrypted EHRs and the encrypted data key; (3) the patients uploads the signed ciphertext to the server; (4) the server sends new PPRs as part of the transactions of the next block; (5) once the new block is mined and verified, the server unlinks the old PPRs and links the new PPRs; (6) the server replaces the old encrypted data key together with the signature with the new ones; (7) the server sends a notification to the patients whether the access policies are successfully modified.

- **Attribute key update** Different from access policy changes, attribute key update will not cause any changes on the blockchain, because the attribute key update will not affect the way how patients’ EHRs are accessed. The whole process goes as follows: (1) the patient downloads the encrypted EHRs from the server; (2) the patient encrypts the data key with new attribute keys, sign the encrypted EHRs and the encrypted data key; (3) the patient uploads the signed ciphertext to the server; (4) the server replaces the old encrypted data key together with the signature with the new ones; (5) the server sends a notification to the patients whether the update is successful.

**User authentication**

Before accessing a patient’s EHRs, a user needs to be authenticated first. PKI-based authentication is one of the mostly used authentication approach. Since the public and private key pair of a user on blockchain is based on one of its many addresses, it will be hard to trace his identity when necessary. In this paper, we suggest using an authentication approach based on attributes (i.e., ABA). There are several different reasons. First of all, it provides anonymous authentication, which means that the user’s identity is still unknown to the verifier except for that the user satisfies the authentication requirement. Secondly, the signature generated for
the authentication can serve as an evidence or be used to trace the user who has requested to access the EHRs. According to [6, 18], a signature generated by the user during authentication can be opened to reveal the user’s identity and thus serve as an evidence or for audit information. Thirdly, in order to access encrypted EHRs, users have to obtain attribute keys and maintain them. Therefore, these keys can also be used for authentication without any extra storage cost. If the authentication and data decryption happen successively, we can furthermore design more efficient protocols that can fulfill the requirements of both authentication and data decryption, to avoid part of the overlapped computations.

Miners and incentive measures

Service providers can act as miners to mine new blocks to hold record changes (contracts changes more specifically) of their users or to pay for the service from attribute issuers if they need to use their services. One interesting but still possible reason for attribute issuers to be miners is that to be a certified attribute issuer, one of the criteria can be that they need to have enough proven contribution to the system, and the contribution is measured by how many blocks they have mined. Except for service providers and attribute issuers, entities such as researchers or other medical organizations can also act as miners to pay for the use of shared data. For the patients, on the one hand, they may need to pay for the service from their service providers or attribute issuers. However, on the other hand, they may get paid by sharing their data. If what a patient earns by sharing his data is not enough to pay for the service, he can also choose to act as a miner if he wishes to.

4 Security discussion

In this section, we discuss the security issues related to the approach we proposed in Section 3. The main goal of the proposed approach is to provide a solution for the main security issues discussed in Section 1, including confidentiality, privacy, access control, integrity, data authenticity, audibility, transparency and interoperability. Based on the approach proposed in Section 2, we discuss each of these security issues in the following:

- **Confidentiality** means not revealing contents to those who are not supposed to know. In this paper, it includes two aspects, the confidentiality of patients’ EHRs and contracts (PPRs) stored on the blockchain. On the one hand, patients’ EHRs are encrypted and can only be accessed by those allowed to, where the details can be found in EHR sharing in Subsection 3. On the other hand, PPRs stored on the blockchain are encrypted and then can be searched by privacy preserving keyword searches.

- **Privacy** provides user anonymity, which is mainly anonymous authentication in this paper and is achieved by pseudo address based public and private key pair, ABE and ABA. Users can create multiple addresses, based on which public and private key pairs are generated to sign, for example, PPRs, EHRs changes and so on. In PKI-based infrastructure, public and private key pairs can normally be considered as users’ identifying information. However, since the public and private key pair is based on one of a user’s many addresses, it is generally hard to identify a user by his key pairs or his traces left on the blockchain. In addition, since PPRs and EHRs are encrypted, sensitive information will also be protected.
• **Access control** There are mainly two aspects related to the access of EHRs, i.e., a patient accessing his own EHRs and a user accessing other patients’ EHRs. A patient accessing his own data is through the interface (as described in Subsection 2 and Fig. 1), by which all his data crumbs are united together as a whole one from the patient’s point of view. However, the process is quite different from a user accessing a patient’s EHRs. First of all, the user searches what he is interested by keywords and the results (usually encrypted PPRs) will be sent to the user. If the user is allowed to access the data, he can execute the PPRs and will be returned with the requested EHRs that are encrypted by ABE (Refer to Subsection 3 for more details).

• **Integrity** Considering the security of hash functions, it is computationally difficult to tamper the content of one block without changing the hash value stored in the next block. Furthermore, every node (or service nodes for better efficiency in practice) has a copy of the blockchain data, the change of a certain block will be easily detected if a node communicates with other nodes. Besides, even though patients’ EHRs are not stored on the blockchain, their merkle roots [4] are included in the block. As a result, if the EHRs are tampered, the merkle root value will change, which will then cause the content change of the block. Therefore, by using blockchain, EHRs can be stored secure and correct without being tampered.

• **Data authenticity** Patients need to sign their EHRs before related contracts are added to the blockchain. Given these signatures as well as the integrity provided by the blockchain, data authenticity can be provided.

• **Audibility and Transparency** Since all changes of patients’ data will be recorded and then added to the blockchain, it is possible to keep a integrate, clear and transparent trail audit of EHR changes, and patients can have access to these trails as long as he has access to the blockchain network. In addition, each user who wants to access EHRs should be authenticated first by ABA and the signature will be stored and can be used later to trace the user’s identity.

• **Interoperability** Unlike a centralized network where different healthcare organizations have their own systems, the decentralization of blockchain is designed to be convenient for each entity to communicate with each other. However, it doesn’t mean that a healthcare system built on blockchain will have no difficulty concerning to interoperability. [12] has proposed that except for validity check, semantic and format check of EHRs should also be required and only those blocks that have passed both checks can be added to the blockchain, which means that all entities of the blockchain should agree on the same standard how EHRs should be expressed. This can also be adopted in our approach to provide interoperability without any extra effort in modifying the main structure.

5 Conclusions

In this paper, we have proposed a blockchain-based approach to build a decentralized healthcare network for secure EHRs sharing. Our main concern is to solve the security issues of confidentiality, access control, privacy, audibility and so
on. By adopting the structure of MedRec, the recording of any EHR changes on
the blockchain keeps patients' data safe from being tampered and enables an easily
accessible, integrate and transparent audit information for the patients. Furthermore,
the application of signcryption provides data authenticity and a flexible way to access
the shared data, while ABA allows a provable trace of users who have requested access
of patients’ EHRs.

References

[1] A secure and efficient ciphertext-policy attribute-based signcryption for personal
health records sharing in cloud computing. *Future Generation Computer Systems*,

478, Berlin, Germany, 2012.

blockchain for medical data access and permission management. In *2016 2nd
International Conference on Open and Big Data (OBD)*, 2016.

2016.


University of Bath, 2009.

[7] Paul Tak Shing Liu. *Medical Record System Using Blockchain, Big Data and

Cryptographers’ Track at the RSA Conference*, pages 376–392, Berlin, Heidelberg,
2011. Springer Berlin Heidelberg.

[9] Miguel Morales-Sandoval, Jose Luis Gonzalez-Compean, Arturo Diaz-Perez, and
Victor J. Sosa-Sosa. A pairing-based cryptographic approach for data security in

*Privacy Preserving Ranked Keyword Search over Encrypted Cloud Data*, pages 396

search over encrypted cloud data with ranking. *Distributed and Parallel Databases*,


