A Decentralizing Attribute-Based Signature for Healthcare Blockchain

You Sun*†, Rui Zhang*†, Xin Wang*†, Kaiqiang Gao§ and Ling Liu‡
*State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China 100093
Email: sunyou,zhangrui,wangxin@iie.ac.cn
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China 100049
§China Electric Power Research Institute, Beijing, China 100192 Email: kaiqiang.gao@gmail.com
‡College of Computing, Georgia Institute of Technology, Atlanta, GA, USA 30332-0765 Email: lingliu@cc.gatech.edu

Abstract—**Blockchain is one of the technology innovations for sharing data across organizations through a peer to peer overlay network. Many blockchain-based data sharing applications, such as sharing Electronic Health Records (EHRs) among different Care Delivery Organizations (CDOs), require privacy preserving verification services with dual capabilities. On one hand, the users want to verify the authenticity of EHR data as well as the identity of the signer. On the other hand, the signer wants to keep his real identity private such that others cannot trace and infer his identity information. However, typical blockchain systems that use pseudonyms as public keys, such as Bitcoin's blockchain, cannot support such privacy-preserving verification. In such systems, it is hard to verify the authenticity of signer's identity, and adversaries or curious parties can guess the real identity from the series of statements and actions taken with a specific pseudonym through inference attacks, such as by transaction graph analysis. In this paper, we propose a decentralized attribute-based signature scheme for healthcare blockchain, which provides efficient privacy-preserving verification of authenticity of EHR data and signer's identity. We also describe a holistic on-chain and off-chain collaborative storage system for efficient storage and verification EHR data. The analysis and experiments show that our scheme is effective and deployable.**

I. Introduction

Blockchain is a distributed general ledger technology and a core enabling capability of digital cryptocurrency Bitcoin [1]. It not only can effectively solve the problem of the Byzantine and dual payments of digital currency, but also breaks away the limitations of traditional centralized leger systems. Instead of relying on trusted third parties, blockchain can establish trust among a network of nodes in a fully decentralized manner through the decentralized verification and consensus mechanism. Blockchain is also considered as an innovative technology framework for establishing a distributed and peer-to-peer trust relationship in many business, science and engineering fields. A typical application is in the healthcare field. By building a consortium blockchain, Electronic Health Records (EHRs) can be easily and safely shared among Care Delivery Organizations (CDOs) with high efficiency and secure data integrity verification.

In the Bitcoin's blockchain [1], a user can generate as many pseudonyms (public and private key pairs) as he wishes. Then he can use any of them to sign transactions. The transactions will be verified using corresponding public key without revealing the real identity of the user. However, this method is not practical for preserving privacy in the healthcare field. First, since the blockchain is publicly available, adversaries or curious parties can guess the real identity from the series of statements and actions taken with a specific pseudonym through inference attacks by transaction graph analysis [23]. Second, in the EHR sharing environment, users need to verify the authenticity of the EHR data and the reliability of its source. Users need to trust that the EHR data is indeed released by a legitimate source.

Attribute-Based Signature (ABS) is proposed in 2011 by Maji, Prabhakaran and Roseulek [14]. We argue that ABE can be leveraged for providing privacy-preserving verification in a distributed blockchain system designed for EHR sharing across CDOs. In an attribute-based signature scheme, the signer's signature key is associated with his series of attributes, so the verifier only knows the signer's attributes and does not know the signer's identity information. This enables users to effective verify the authenticity of an EHR data record while protecting the real identity of the signer.

However, to employ the attribute based signature (ABS) scheme in a healthcare blockchain, which is a decentralized network system, we need to address two important technical challenges. First, attribute certificates are issued from different authority agencies in reality in existing ABS schemes, and thus, a central authority agency is assumed to establish trust, supervise and generate global parameters. Therefore, how to realize EHR data security sharing across different CDO systems without relying on a central authority agency is the first open issue. Second, the storage capacity of the existing blockchain system is limited. If all EHR data stored in each CDO are moved to and stored on the blockchain maintained by the network of EHR users, it will result in a huge storage and computation burden for the blockchain system, especially when the network size of such a blockchain system grows. Therefore, how to efficiently and securely store EHR data while disperse the storage and computation burden for a distributed blockchain system presents another open issue that demands effective solutions.

In this paper, we present a holistic approach to address both

978-1-5386-5156-8/18/$31.00 ©2018 IEEE

challenges with three original contributions.

- First, we propose a decentralizing attribute-based signature (called DABS) scheme for providing privacy preserving verification service in a healthcare blockchain. The scheme has two salient features: (1) It can effectively verify the attributes of the signer without exposing the identity of the signer; (2) The decentralized attribute based signing property makes it suitable for the distributed blockchain system. In our DABS scheme, multiple attribute authorities may issue attribute certification and corresponding signature keys to users without relying on a central authority agency to supervise and manage them.
- Second, we propose a blockchain-based EHR data storage system for secure sharing EHR data among different CDOs though an effective on-chain and off-chain collaboration storage model. Our blockchain based storage system has a number of advantages: (1) Using blockchain to realize security sharing of EHR data across different CDOs such that the stored and shared EHR data are non-tampered, unforgeable and verifiable. (2) We adopt the combination of on-chain and off-chain storage to realize the secure sharing of large-scale and distributed EHR data. The address of each EHR data record is stored in a transaction on the blockchain, and the EHR data is stored in each node off the blockchain. This makes it easier for users to locate each piece of EHR data while circumvents the storage limitation of the blocks.
- Finally, We provide formal security analysis of the proposed DABS verification protocol with respect to unforgeability, security against collusion attacks, anonymity, and non-repudiation. Our experimental evaluation demonstrates that the proposed DABS scheme is effective and easily deployable.

The rest of the paper is organized as follow. Section II gives a review of the related work with respect to attribute-based cryptography and applications on healthcare blockchain. Preliminaries of our protocol including linear secret-sharing schemes (LSSS) and composite order bilinear groups are provided in Section III. In Section IV, we present a decentralized attribute-based signature algorithm for privacy-preserving verification. In Section V, we apply the proposed DABS scheme to the healthcare blockchain and describe the protocol in detail. Section VI gives the security analysis of our protocol. In Section VII, we show the performance of our protocol. Finally, we conclude the work in Section VIII.

II. Related Work

In this section, we give the discussion on attribute-based cryptography and the applications on blockchain systems.

A. Attribute-Based Cryptography

Attribute-based cryptography can automatically implement fine-grained access control through cryptographic algorithms, while without revealing users' identity information.

Attribute-based encryption (ABE) is a public-key cryptographic algorithm whose user keys or ciphertexts are related to attributes. In this type of cryptosystem, the ciphertext can be decrypted only when the attribute corresponding to the user key satisfies the ciphertext corresponding to the attribute. The concept of ABE was first proposed by Amit and Brent [4], then Vipul et al. made a further definition of it [5]. The ABE scheme can be divided into two types: Key-Policy Attribute-Based Encryption (KP-ABE) [5] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [6]. In KP-ABE, the generation of user key is based on a predefined access tree, and the ciphertext is associated with a set of attributes. The KP-ABE is mainly aimed at the storage of sensitive information of users in third parties, and it is difficult to control the access rights flexibly. This scheme can prevent the user from directly telling the access key to the authorized user. Instead, it authorizes the user to recover the key according to their own attributes. This key can only decrypt the data that he has permission to access. Access control of encrypted data is essentially achieved through the access control of the key. In CP-ABE, using access tree to encrypt data, the user key is associated with a set of attributes. That is, the access structure is defined in the ciphertext. Only users who satisfy the access structure can decrypt it without using a trusted server to achieve this control, which is conceptually closer to traditional access control methods. In recent years, Researchers conducted further research on ABE mainly in two aspects: multi-centric joint generation of user keys [7]–[11] and enabling it to support arbitrary predicates [12], [13].

With the development of ABE, Attribute-Based Signature (ABS) [14], [15] schemes have also been proposed. In the ABS scheme, the signer uses the private key associated with the attribute to sign the message. The verifier only knows the attribute of the signer and does not know its true identity information, thereby realizing the protection of the signer's identity information.

The current attribute-based signature schemes can be divided into two types. The first type is based on a linear threshold structure, and the second type is based on the access control tree of attribute. The schemes based on a linear threshold structure originated from fuzzy identities-based signature, and the set of attributes between signature and verification should have a certain overlap, which is also called a threshold. Such schemes mainly include the ring signature schemes and the threshold signature schemes.

The signature scheme based on the access control tree includes the Guo-Zeng signature scheme [17] and the group signature scheme. The private key corresponding the attribute in the Guo-Zeng scheme is associated with the access structure defined by the verifier, and one signature can be verified if and only if the attribute of private key corresponding to the signature matches the attribute that need to be certified. The group signature scheme is represented by Khader's attribute-based group signature [18]. The private key is independent of the access structure defined by the verifier. Attribute-based signature is widely used in anonymous credential systems [19]. In this kind of systems, users have their own pseudonym, and organizations issue credentials to the user by their pseudonym.

For different organizations, the same user may have different pseudonyms. The user needs to prove that he has a certain pseudonym and has obtained the organization's certificate. It can achieve fine-grained access control but it cannot support complex predicates. Document [15] proposes an ABE-based access control system with non-transferability that solved the issue of certificate sharing.

ABS can provide effective verification of signer's attributes while without revealing signer's identity information. However, in many applications the predefined attributes might issued from different organizations and domains. For instance, a party might want to verify the shared EHR data signed by a user who has the attribute of "doctor" issued by a medical organization and the attribute "researcher" issued by the administrators of a clinical trial. Using existing ABS schemes for this application can be problematic since one needs a single authority that is both able to verify attributes across different organizations and issue private keys to every user in the system.

Fortunately, Allison and Brent proposed the Distributed Attribute-Based Encryption (DABE) scheme [11] for issuing attributes and private keys cross organizations. In DABE, the system does not need a fixed attribute authority, but there can be multiple attribute authorities in the distributed network to complete the issue of attribute certificates. This scheme makes it possible to apply ABS to a blockchain system.

B. Healthcare Blockchain

In recent years, blockchain has attracted attention of academia and industry. It is used in many fields to realize distributed resources storage and sharing. A typical application is in the healthcare field to achieve EHR data sharing cross different CDOs.

Document [20] uses blockchain to guarantee the authenticity of medical data and prevent data from being altered by malicious attackers. The MedRec framework [21], [22] realizes automatic privilege management, combines smart contract with access control, and integrates distributed medical data and privilege management of different organizations. But the framework uses the proof of work (PoW) consensus mechanism, and the computation cost to maintain blockchain consistency is very huge.

In this paper, we propose a blockchain-based EHR sharing protocol. Inspired by DABE scheme [11], we use a DABS scheme to replace Elliptic Curve Digital Signature Algorithm (ECDSA) that used in Bitcoin [1] to achieve attribute verification and privacy-preserving of signer's identity information on the healthcare blockchain environment.

III. Preliminaries

In this part we simply explain the basic preliminaries including LSSS and composite order bilinear groups used in our constructions.

A. Linear Secret-Sharing Schemes

The secret sharing scheme is to split the secret in an appropriate manner. Each share after splitting is managed by different participants. A single participant cannot recover secret information. Only a few participants can cooperate to recover secret information. Secrets can still be fully recovered when there is a problem with participants in any of the corresponding areas.

A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called linear if

- The shares for each party form a vector over $\mathbb{Z}_p$.
- There exists a $k \times m$ share-generating matrix A for Π. For all x from 1 to k, the x^{th} row of A is labeled by a party $\rho(x)$. The column vector $v = (s, r_2, ..., r_m)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, ..., r_m \in \mathbb{Z}_p$ are randomly chosen, then Av is the vector of k shares of the secret s according to Π. The share $(Av)_x$ belongs to party $\rho(x)$.

Suppose that Π is an LSSS for A, let S be an authorized set, $I = \{x \mid \rho(x) \in S\}$ represent $I \subseteq 1, ..., k$. Then there exist constants $\{\omega_x \in \mathbb{Z}_p\}_{x \in I}$ such that, for any valid shares $\{\lambda\}_x$ of a secret s according to Π, we have $\Sigma_{x \in I} \omega_x \lambda_x = s$. These constants $\{\omega_x\}$ can be found in polynomial time with respect to the size of the share-generating matrix A.

B. Composite Order Bilinear Groups

In our scheme, we use the composite order bilinear groups. Define a group generator $\mathcal{G}$ and a security parameter λ, an algorithm which takes λ as input and outputs a description of a bilinear group G. We will have $\mathcal{G}$ output $(p_1, p_2, p_3, G, G_T, e)$ wherep_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1 p_2 p_3$,and $e : G^2 \rightarrow G_T$ is a map such that:

- $\forall g, h \in G, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{a,b}$
- $\exists g \in G$ such that $e(g, g)$ has order n in G_T

Assume that the group operations in G and G_T as well as the bilinear map e are computable in polynomial time, the group descriptions of G and G_T include generators of the respective cyclic groups. G_{p_1}, G_{p_2}, and G_{p_3} is the subgroups of order p_1, p_2, p_3 in G. When $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, $e(h_i, h_j)$ is the identity element in G_T. Suppose $h_1 \in G_{p_1}$ and $h_2 \in G_{p_2}$. g is a generator of G. Then, $g^{p_1 p_2}$ generates G_{p_3}, $g^{p_1 p_3}$ generates G_{p_2}, and $g^{p_2 p_3}$ generates G_{p_1}, so for some $\alpha_1, \alpha_2, h_1 = (g^{p_2 p_3})^{\alpha_1}$ and $h_2 = (g^{p_1 p_3})^{\alpha_2}$. Then $e(h_1, h_2) = e(g^{p_2 p_3 \alpha_1}, g^{p_1 p_3 \alpha_2}) = e(g^{\alpha_1}, g^{p_2 p_3})^{p_1 p_2 p_3} = 1$.

IV. Decentralizing Attribute-Based Signature

In this section, we propose a Decentralizing Attribute-Based Signature (DABS) algorithm. In our DABS scheme, users are issued attributes and private keys from multiple authorities belong to different organizations. Every user in the system has a globally verifiable identifier GID, which can bind keys issued by different authority to the same user.

The construction of DABS algorithm is shown as below:

Global Setup$(\lambda) \rightarrow GP$ At this stage, the system generates a bilinear group G of order N through the input security parameters λ, which generator is g. The global public parameter GP of the system consists of N and g, which is

$GP = (N, g)$. Choose a hash function H which maps the global identities GID to an elements of G.

Authority Setup$(GP) \rightarrow SIK, VK$ Each authority runs the authority setup algorithm with the input of the global public parameter GP, and it generates a corresponding signature key SIK and verification key VK for each attribute i.

First, the algorithm randomly chooses $\alpha_i, y_i \in \mathbb{Z}_N$ for each attribute i. Then calculate the verification key $VK = \{e(g,g)^{\alpha_i}, g^{y_i} \forall i\}$, and the signature key $SIK = \{\alpha_i, y_i \forall i\ \}$.

KeyGen$(GP, GID, i, SIK) \rightarrow SIK_{i,GID}$ The **KenGen** algorithm generates the signature key for each attribute i corresponding to each GID. The authority can use this algorithm to calculate: $SIK_{i,GID} = \{g^{\alpha_i}, H(GID)^{y_i} \forall i\}$.

Sig$(GP, M, (A, \rho), \{SIK_{i,GID}\}) \rightarrow \sigma$ This algorithm takes the global public parameter GP, the message M, the access control matrix A with ρ maps each row of matrix A to an attribute, and the signature keys of the authority as input, and it outputs the signature σ. The size of access matrix A is $n \times l$. Randomly select $s \in \mathbb{Z}_N$ and a vector $v \in \mathbb{Z}_N^l$. The first element of the vector v is s. Define A_x as the row x of matrix A, and compute $\lambda_x = A_x \cdot v$. Then randomly choose a vector $w \in \mathbb{Z}_N^l$, and the first element of w is 0. Compute $\omega_x = A_x \cdot w$. Select a random $r_x \in \mathbb{Z}_N$ for each A_x. Choose a hash function $H^{'}$, then calculate:

$$Sig_0 = e(g,g)^{sH^{'}(M)},$$

$$Sig_{1,x} = H(GID)^{r_x},$$

$$Sig_{2,x} = \frac{e(g,g)^{\lambda_x} \cdot e(H(GID), g^{\omega_x})}{e(g^{\alpha_{\rho(A_x)}}, g) \cdot e(H(GID)^{y_{\rho(A_x)}}, g^{r_x})}.$$

The signature $\sigma = (Sig_0, Sig_{1,x}, Sig_{2,x})$.

Ver$(GP, M, \sigma, \{VK\}, (A, \rho)) \rightarrow \{0, 1\}$ The **Ver** algorithm calculates c_x by access matrix A, so that

$$\sum_x c_x A_x = (1, 0, ..., 0).$$

Then use the hash function $H^{'}$ to get the hash of the message $H^{'}(M)$. Verify whether the equation $Sig_0^{\frac{1}{H^{'}(M)}} = \prod_x (e(g,g)^{\alpha_{\rho(A_x)}} \cdot e(Sig_{1,x}, g^{y_{\rho(A_x)}}) \cdot Sig_{2,x})^{c_x}$ is established. If the equation holds, it is indicated that the signature comes from the user that satisfies the attributes of the access matrix A, the algorithm output 1. Otherwise, reject the signature and return $\perp$.

Correctness. Assuming a user's global identity is GID, and he has a series of attributes that make up the access matrix A. He signed the message M to get the signature $\sigma = (Sig_0, Sig_{1,x}, Sig_{2,x})$. Next we will prove that (M, σ) can be verified by the **Ver** algorithm.

It can compute that:

$$\begin{aligned}
&\prod_x (e(g,g)^{\alpha_{\rho(A_x)}} \cdot e(Sig_{1,x}, g^{y_{\rho(A_x)}}) \cdot Sig_{2,x})^{c_x} \\
&= \prod_x (e(g,g)^{\alpha_{\rho(A_x)}} \cdot e(H(GID)^{r_x}, g^{y_{\rho(A_x)}}) \cdot \\
&\quad \frac{e(g,g)^{\lambda_x} \cdot e(H(GID), g^{\omega_x})}{e(g^{\alpha_{\rho(A_x)}}, g) \cdot e(H(GID)^{y_{\rho(A_x)}}, g^{r_x})})^{c_x} \\
&= \prod_x (e(g,g)^{\lambda_x} \cdot e(H(GID), g)^{\omega_x})^{c_x}
\end{aligned}$$

Since $\lambda_x = A_x \cdot v$, $\omega_x = A_x \cdot w$ and $v \cdot (1, 0, ..., 0) = s$, $w \cdot (1, 0, ..., 0) = 0$, therefore

$$\prod_x (e(g,g)^{\lambda_x} \cdot e(H(GID), g)^{\omega_x})^{c_x} = e(g,g)^s = Sig_0^{\frac{1}{H^{'}(M)}}$$

The verification passed.

V. Blockchain-Based EHR Sharing Protocol

In this section, we describe our blockchain-based EHR sharing protocol that combining DABS scheme to achieve privacy-preserving verification. We also build an on-chain and off-chain EHR storage model for efficiency storing.

A. System Model

In this sbusection, we introduce the system model of our blockchain-based EHR sharing protocol, including the storage model, roles, nodes, data structure and consensus mechanism.

1) On-Chain and Off-Chain Storage Model: For the problem of limited storage capacity and computational resources of the blockchain, we adopts a combination of blockchain and off-chain storage to implement data storage. That is, only the addresses of the EHR data stored on the blockchain, and the EHR data is encrypted and stored in each nodes. This makes more easier to sharing EHR data cross different CDOs while avoiding cumbersome data migration.

Specifically, as shown in Fig. 1, once a doctor has created an EHR data, he signs the EHR data with his private keys related with his attributes and stores the signed EHR data in the his own database. Then the doctor can share the EHR data by signing the address of the stored data with his attributes and publishing it on the blockchain. When users want to access the EHR data, he first verify the publisher's signature of the stored address on the blockchain, and then retrieve the EHR data from the node and verify the EHR data.

2) Roles: There are mainly three roles in our protocol: user, authority agency and administrator.

User: in our protocol, users includes doctor, patient and other users such as researcher. Doctors are responsible for creating EHR data and signing the data with their own attributes. Doctors also can share the EHR data with other users by broadcast its address in the blockchain. Patient and other users can retrieve and access the EHR data whose stored address is published on the blockchain by verifying the signatures of both EHR data and its address.

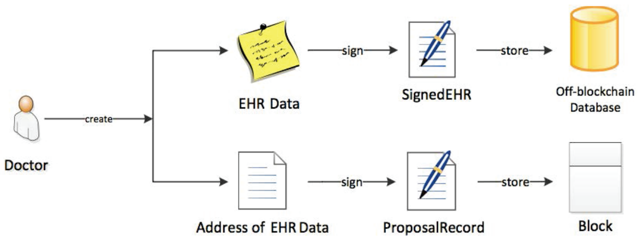

Fig. 1. On-Chain and Off-chain Storage

Authority Agency: The different attributes of the user are issued by one or more authority agency. Authority agencies are responsible for issuing the signature keys related with the attributes to the users.

Administrator: the administrator generates a global public parameter GP when the system is initialized, and it assigns a global identity GID to each user entering the system. Administrator also manages EHR data.

3) Nodes: In our protocol, nodes are divided into two categories: primary node and backup node.

Primary Node: They collect a group of transactions broadcasted on the network into one block, creating a new block.

Backup Node: They can create new transactions and publish them to the network. If they satisfy the access policy, they can verify the signature of the transaction.

In our scheme, in addition to having its own unique identifier, a node has a series of attributes. Every transaction that the user publishes to the block will carry his signature. This signature does not reveal the identity of the signer, but is based on a series of attributes of the signer. When a user accesses data in a block, he needs to verify this signature first. Verification can pass only if the signature matches a specific attribute. Conversely, if the signer's attributes do not meet the verifier's requirements, the verification fails, indicating that this is not an EHR created by a doctor who meets these specific requirements. Attribute-based signatures can not only achieve the anonymity of signers, but also can effectively verify the authenticity of EHR data.

4) Data Structure: In the Bitcoin system [1], a block includes a block header and a main block. The block header is made up of a version number, a previous block hash, a current block hash, timestamp, nonce and so on. The main block consists of a series of transactions. Similarly, our healthcare blockchain also has block header and main block in each block.

In our Healthcare Blockchain, there are four kinds of data structure: SignedEHR, ProposalRecord, BlockHeader and MainBlock, we will introduce these four data structures in detail.

SignedEHR: The SignedEHR is stored in a trusted third-party database off-blockchain. A doctor creates EHR data and encrypt it, then signs it with his attribute-based signature. The data structure of SignedEHR is shown as below:

$$SignedEHR = (EHR, \{VK_i\}, Sig_{SIK}(EHR, timestamp))$$

where Sig is our DABS algorithm, $\{VK_i\}$ is a series of verification keys corresponding to the attributes i, SIK is the doctor's signature key based on his series of attributes.

ProposalRecord: The ProposalRecord transactions in the blockchain mainly stored the corresponding address of the SignedEHR, and signed by doctors with DABS algorithm. Its structure is shown as below:

$$ProposalRecord = (Addr, \{VK_i\}, Sig_{SIK}(Addr, timestamp))$$

where Sig is our DABS algorithm, SIK is the doctor's signature key based on his series of attributes. $Addr$ is the address of the SignedEHR data, and it also needs a timestamp, and $\{VK_i\}$ is a series of verification keys corresponding to the attributes i.

BlockHeader: The BlockHeader contains a current block hash, a previous block hash, a timestamp and a signature :

$$BlockHeader_n = (H_{Block_n}, H_{Block_{n-1}}, timestamp, Sig_{SIK})$$

where H is the hash function, Sig_{SIK} is the signature of the block publisher.

MainBlock: A MainBlock is a collection of ProposalRecord transactions, which supports both present and future care that patients receive from the same or other clinicians or care providers. Storing EHR data on the blockchain can facilitate the sharing of EHR data among different CDOs. Primary nodes integrate a series of ProposalRecord on the network into one block. The structure of the block is as follows:

$$MainBlock = \{ProposalRecord_1, ..., ProposalRecord_n\}$$

5) Consensus Mechanism: In our system, we use Practical Byzantine Fault Tolerance (PBFT) [3] as the consensus mechanism. PBFT algorithm was proposed by Miguel Castro and Barbara Liskov in 1999. The original Byzantine fault tolerant algorithm is inefficiency and PBFT solves this problem, making the Byzantine fault tolerant algorithm feasible in practical system application. It is mainly divided into three phases: pre-prepare, prepare, and commit.

In this process, U express the collection of all replicates and each replicate is represented as an integer of 0 to $|U|-1$. If there are at most f replicates may fail, we assume that $|U| = 3f - 1$, where $|U|$ is the number of replicate sets. The responsibility of the primary is to generate new blocks, and it is elected from all nodes. The formula $p = view\ mod\ |U|$ gives the main node, where $view$ is the view number and p is the replicate number.

In the pre-prepare phase, the primary node assigns a sequence number seq to the received request, and then sends a

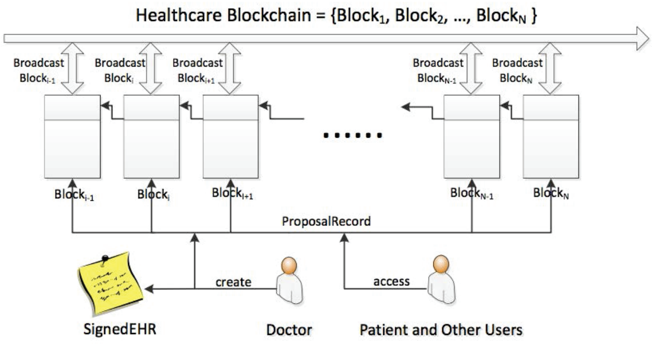

Fig. 2. The Flow of Our Protocol.

pre-prepare message to all backup nodes in one view $view$. The format of the pre-prepare message is as follows:

$$\langle\langle PRE - PREPARE, view, seq, d\rangle, M\rangle$$

where d is the hash value of M.

In the next phase, node i send the prepare message:

$$\langle PREPARE, view, seq, d, i\rangle$$

to all replica nodes. Then node i writes the pre-prepare and prepare messages to its message log.

In the last phase, if a node receive $2f$ message that have the same hash value as itself, replicate i broadcast:

$$\langle COMMIT, view, seq, D(M), i\rangle$$

After that, each replicate should check the signature, view number and sequence number of the message. Finally, if a node receives $2f + 1$ commit messages, it submits the new block to the healthcare blockchain.

B. The Proposed Protocol

In the healthcare blockchain, only doctors can create EHR data and put the address of EHR data on the blockchain. After the doctor creates an EHR data, it signs the data and broadcasts the address of it to the blockchain with his signature. The signed EHR data is stored in an off-chain database. Patients and other users in our scheme are verifiers. They can only access the data on the blockchain, and verify the signer's attributes. They cannot broadcast EHR-related data to the blockchain. This process is shown in Fig. 2.

Our scheme mainly contains five phases: global setup phase, authority setup phase, user setup phase, proposal phase and verification phase.

1) Global Setup: Global setup phase is to install the Healthcare scheme. The input of this phase is a security parameters λ, and the global setup algorithmic will generate a bilinear group G with the generator g, and its order is N. The algorithm output the global public parameter $GP = (N, g)$. In addition, it will choose a hash function that will be used for the system. This hash function H can maps the global identities GID to an elements of G.

The Administrator runs this algorithm at the stage of system initialization.

Algorithm 1 Global Setup Phase

```
Input: a security parameters $\lambda$.
Output: a bilinear group $G$ of order $N$, which generator is
    $g$, the global public parameter $GP = (N, g)$, and a hash
    function H which maps the global identities $GID$ to an
    elements of $G$.
```

2) Authority Setup: In our scheme, each authority can issue users the signature keys corresponding to their attributes, and every authority needs to run this authority setup algorithm. The authority setup phase takes global public parameter $GP = (N, g)$ as input, and output the signature key SIK and the verification key VK for each attribute i. Each SIK and VK for attribute i requires the signature of the authority that issued it.

Through this phase, it is randomly selected $\alpha_i, y_i \in \mathbb{Z}_N$ for each attribute i, and then calculate the verification key $VK = \{e(g, g)^{\alpha_i}, g^{y_i}\}$ and the signature key $SIK = \{\alpha_i, y_i\}$ for each attribute i.

Algorithm 2 Authority Setup Phase

```
Input: global public parameter $GP$.
Output: the signature key $SIK$ and the verification key $VK$
    for each attribute $i$.
```

3) User Setup: Every doctor and patient entering the system needs to run this algorithm. The user needs to submit an application to the authority, and the authority will then issue the signature key corresponding to the user's attribute and some system parameters to be used. It takes the global public parameter $GP = (N, g)$, user's global identities GID, a set of the user's attributes i and the signature key $SIK = \{\alpha_i, y_i\}$ as input, and generate the signature key set for his each attribute.

In this phase, it calculate the user's signature key $SIK_{i,GID} = \{g^{\alpha_i}, H(GID)^{y_i}\}$ for each attribute i.

Algorithm 3 User Setup Phase

```
Input: the global public parameter $GP$, the global identities
    of the user $GID$, a set of the user's attribute $i$ and the
    signature key $SIK$.
Output: the signature key set $\{SIK_{i,GID}\}$ for each attribute
    $i$ corresponding to each $GID$.
```

4) Proposal: The proposal phase is divided into two parts, first, the doctor generates EHR data, signs the data, and stores it in the off-blockchain database. Then, the address of the EHR data is signed and published as a transaction on the blockchain.

The doctor with a global identity GID constructs the access control matrix A based on his attributes and the map ρ, where each row of the access control matrix A represents a user's attribute i, A is a $n \times l$ matrix.Let A_x as the row x of matrix A. Choose a random number $s \in \mathbb{Z}_n$ and a vector $v \in \mathbb{Z}_n^l$, let s be the first element of v. For each x, calculate $\lambda_x = A_x \cdot v$. Randomly select a vector $w \in \mathbb{Z}_n^l$ with the first element 0, calculate $\omega_x = A_x \cdot w$ for each x. For each row A_x of matrix A, randomly choose $r_x \in \mathbb{Z}_n$, select a hash function $H^{'}$.

Then the user can sign the message M with the $SIK_{i,GID} = \{g^{\alpha_{\rho(A_x)}}, H(GID)^{y_{\rho(A_x)}}\}$ for each row x of A, the signature is divided into three parts: $\sigma = (Sig_0, Sig_{1,x}, Sig_{2,x})$. where $Sig_0 = e(g,g)^{sH^{'}(M)}, Sig_{1,x} = H(GID)^{r_x}, Sig_{2,x} = \frac{e(g,g)^{\lambda_x} \cdot e(H(GID), g^{\omega_x})}{e(g^{\alpha_{\rho(A_x)}}, g) \cdot e(H(GID)^{y_{\rho(A_x)}}, g^{r_x})}$. The user's signature process for EHR address is the same as the signature process for EHR data.

```
Algorithm 4 Proposal Phase
Input: the global public parameter $GP$, the EHR data created by the doctor $M$, an access control matrix $A$ with $\rho$ maps each row of matrix $A$ to an attribute, and the signature key corresponding to each attribute of the user $\{SIK_{i,GID}\}$.
Output: the EHR data with the doctor's signature $SignedEHR$ and the address of EHR data with the doctor's signature $ProposalRecord$.
  the doctor create the EHR data $M$;
  add a timestamp;
  create an access control matrix $A$ with a map $\rho$;
  calculate the signature of EHR data $M$ and the timestamp based on access control matrix $A$ using the signature key set $SIK_{i,GID}$;
  store the EHR with the signature to the off-blockchain database;
  get the address $Addr$ of the EHR data;
  add a timestamp;
  calculate the signature of EHR address $Addr$ and the timestamp based on access control matrix $A$ using the signature key set $SIK_{i,GID}$;
  return $ProposalRecord$;
```

In the proposal phase, in addition to backup nodes adding transactions to a block, primary nodes also generate new blocks. We use Practical Byzantine Fault Tolerance (PBFT) consensus mechanism [3] in our scheme and each new block will have the generator's signature.

5) Verification: In our system, doctors and patients put the EHR on healthcare blockchain to allow the doctors in different hospitals to view the data. When viewing data, the user needs to verify that the data was issued by a doctor with specific attributes. First the user needs to verify the EHR address stored in the transaction in the blockchain, as it shown in Algorithm 5, then the EHR data stored in the database is found by address and the EHR data is verified, Algorithm 6 describes this process.

When verifying, the user first needs to construct the access control matrix A according to the attribute set and the map ρ. Similarly, A_x represents the row x of matrix A, calculate c_x so that $\sum_x c_x A_x = (1,0,...,0)$. Then get the hash value $H^{'}(M)$ of the message M. Through calculation, verify whether the equation $Sig_0^{\frac{1}{H^{'}(M)}} = \prod_x (e(g,g)^{\alpha_{\rho(A_x)}} \cdot e(Sig_{1,x}, g^{y_{\rho(A_x)}}) \cdot Sig_{2,x})^{c_x}$ is valid using VK in the ProposalRecord. If it is, the verification is passed, otherwise the verification fails.

```
Algorithm 5 Transaction Verification
Input: the global public parameter $GP$, the EHR address $Addr$ and the signature $\sigma$ in $ProposalRecord$, a series of attributes that need to be verified and the corresponding verification key $\{VK\}$.
Output: if the verification is successful, output 1, otherwise output $\perp$.
  use the map $\rho$ create the access control matrix $A$;
  use the hash function $H^{'}$ to get the hash of the address $H^{'}(Addr)$;
  calculate the verification formula with the verification key $\{VK\}$;
  check whether the equation holds?
  return {0, 1};
```

```
Algorithm 6 EHR Data Verification
Input: the global public parameter $GP$, the EHR data $M$ and the signature $\sigma$ in $SignedEHR$, a series of attributes that need to be verified and the corresponding verification key $\{VK\}$.
Output: if the verification is successful, output 1, otherwise output $\perp$.
  find the EHR data in the database according to the address $Addr$;
  use the map $\rho$ create the access control matrix $A$;
  use the hash function $H^{'}$ to get the hash of the message $H^{'}(M)$;
  calculate the verification formula with the verification key $\{VK\}$;
  check whether the equation holds?
  return {0, 1};
```

VI. Security Analysis

In this section, we will discuss the security of our Decentralizing Attribute-Based Signature algorithm from the aspects of unforgeability and resistance to collusion attacks first. Next, we will discuss the security of the blockchain that combines DABS from both anonymity and non-repudiation.

A. Unforgeability

In the DABS scheme, a user who satisfies the access structure can calculate the signature of the message M through the signature key set corresponding to its attributes. It is computationally infeasible that the user who does not have the access structure wants to forge one such signature.

If a user does not have the attribute i and he wants to fake a signature with this attribute. The user does not have the $SIK_{i,GID}$, and when he compute the signature corresponding to the row A_x of the access matrix A with attribute i, he can not calculate the value of $e(g^{\alpha_{\rho(A_x)}}, g)$ and $e(H(GID)^{y_{\rho(A_x)}}, g^{r_x})$. Thus it is not possible to forge the signature. In the case of an adversary may misappropriate user's signature, then use a part of the signature to forge the signature of a new message, assume that the adversary want to use the user's partial signature $Sig_{1,x}$ and $Sig_{2,x}$ to sign a new message M'. The adversary only needs to forge Sig_0. But in our scheme, the adversary can not get the random number s, so it cannot be successfully forged.

B. Security under Collusion Attack

In our DABS scheme, the globally verifiable identifier GID binds a user's private keys from different authority. For the collusion attack of users in the system, assume that there are two users whose globally verifiable identifier are GID_1 and GID_2.The attribute sets they own are S_1 and S_2 respectively, and they want to merge keys to forge a signature of attribute set S, where $S \in S_1 \cup S_2$. Then they need to compute $e(H(GID)^{y_i}, g^{r_i})$ for each $i \in S$. But the user with GID_1 can get $e(H(GID_1)^{y_i}, g^{r_i})$ for each $i \in S_1$ and the user with GID_2 can get $e(H(GID_2)^{y_i}, g^{r_i})$ for each $i \in S_2$. This will cause them being unable to get the signature on the attribute set S. As a result, the collusion attack cannot be completed.

Another situation is the collusion attack of the authority. Suppose the user's attributes come from m authorities, when m authorities jointly launch an attack, they can forge the signature of an access structure with these attributes. However, when the number of authorities that initiate the collusion attack is less than or equal to $m-1$, the signature cannot be forged.

C. Anonymity

In this scheme, we use attributes to identify users without using the user's real identity, this makes the system anonymity. When users enter the system, they will be assigned a series of attributes that can be issued by different authorities and tied together with their global identity GID. They obtain the corresponding signature key for each attribute from the authorities, after that, when they create a new message, they need to sign the message with the key corresponding to the attribute. When other users verify the signature, only the verification key of the attributes corresponding to the signature can be successfully verified. This achieves that the verifier does not know the true identity of the signer, but can know the attributes that the signer has. In other words, after the doctor uploads the EHR data into the healthcare blockchain, when other hospitals or other doctors need to view the data, they do not know the true identity of this doctor, but they can verify whether the EHR data was created by a qualified doctor.

D. Non-Repudiation

Each block and transaction in the scheme will have a signature, which achieves non-repudiation of the system. When a user views a piece of EHR data, he first needs to verify the signature of the block where the transaction is located, this signature is signed by the primary node when the block is generated. In addition, EHR data and EHR addresses created by backup nodes are required to carry their signatures. After the signature verification of the block is successful, the user needs to verify the signature of the EHR address stored in the blockchain, then finds the EHR data in the off-blockchain database according to the EHR address, and also needs to verify the signature of the EHR data. The scheme guarantees the integrity of the information transmission, the identity authentication of the sender, and the prevention of repudiation in the transaction through the signature.

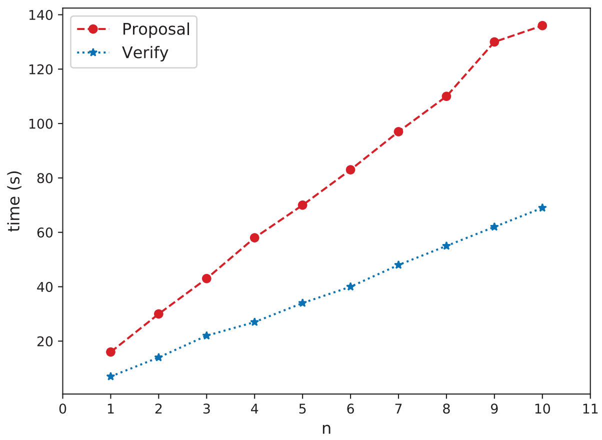

Fig. 3. Computation Overhead of Proposal and Verification

VII. Evaluation

For our prototype, we built an experimental network consists of ten peer nodes and one manage node as administrator and authority agency. We implemented our protocol in C and ran the software implementation on a client machine with Intel i7-4600U 2.70GHz CPU, and 4GB RAM.

Figure 3 shows the computation overhead of proposal and verification phase. The proposal phase mainly contains two steps: signing the EHR data and its storage address. Let n be the number of the attributes required in the process of signature. We vary n from 1 to 10. We have observed that it takes about 16 seconds to generate and sign a transaction with only one attribute. When the number of attributes is increased to 10, the time cost of generating and signing a transaction is 137 seconds. In the verification phase, the time cost of verifying a transaction also contains two parts: the time costs of verifying an EHR data and its storage address. When there is only one attribute, the time cost of verification is about 7 seconds. And when the number of attributes increases to ten, it takes about 68 seconds to verify a transaction. One block may contain more than one transaction (EHR data). The time costs of proposal and verifying a block depend on the number of transactions contained in a block.

From the preceding experiment results, we can see that our protocol is effective and its efficiency is acceptable in a consortium blockchain such as healthcare blockchain. Our codes are not optimised for a particular sparse prime number which might result in very specific and optimised modular reduction, thus the efficiency can be greatly improved by program optimization and implement on GPU.

VIII. Conclusion

Secure sharing of EHR data in a healthcare block chain is an attractive and low cost solution that holds huge potential for the healthcare delivery industry. The capability of verifying the authenticity of EHR data and the identity of the signer for EHR data is critical for practical deployment of a healthcare blockchain. We have proposed a Decentralized Attribute-based Signature (DABS) scheme for providing privacy-preserving verification in a healthcare blockchain system, with two novel features: (1) DABS can effectively verify the attributes of the signer without exposing its identity and (2) the decentralized signing capability makes DABS effective for deployment in a distributed blockchain system. To ensure efficient storage and verification, we also developed an effective on-chain and off-chain collaboration storage model for secure sharing EHR data among different CDOs. Our blockchain based storage system ensures that (1) the stored and shared EHR data are non-tampered, unforgeable and verifiable, and the combination of on-chain and off-chain storage effectively realizes the secure sharing of large-scale distributed EHR data. Finally, we have analyzed the security properties and the performance of the proposed protocol. Our DABS scheme has better security properties with respect to unforgeability and resilience under collusion attacks. The experimental results show that our protocol is effective and practical. Our ongoing research continues along several dimensions. First, we plan to evaluate our prototype system at a larger scale to provide facilities that can ease the practical deployment of DABS and the on-chain and off-chain collaboration storage model. Second, we continue the investigation on making DABS scheme more robust and scalable with respect to the scale of the blockchain, the network size and the emerging threats.

Acknowledgment

The authors from Institute of Information Engineering, Chinese Academy of Sciences acknowledge the partial support from National Key R&D Program of China under Grant No.2017YFB1400700 and National Natural Science Foundation of China under Grant No.: 61472414, 61772514, 61602061. The author from Georgia Institute of Technology acknowledges the partial support from USA National Science Foundation under Grants CNS-1115375, NSF 1547102, SaTC 1564097 and a faculty award from IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. www.bitcoin.org, 2008.

[2] ANSI. ISO/TS 18308 Health Informatics-Requirements for an Electronic Health Record Architecture. ISO, 2003.

[3] Miguel Castro, Barbara Liskov. Practical Byzantine Fault Tolerance. USENIX Technical Program - OSDI 99, 1999.

[4] Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In theory and application of cryptographic techniques, 457-473.

[5] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06). 2006, 89-98.

[6] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07). 321-334.

[7] Melissa Chase. Multi-authority Attribute Based Encryption. In Theory of Cryptography Conference, 2007, 515-534.

[8] Melissa Chase and Sherman S.M. Chow. Improving Privacy and Security in Multi-authority Attribute-based Encryption. In Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09), 121-130.

[9] Taeho Jung, Xiangyang Li, Zhiguo Wan, and Meng Wan. Privacy preserving cloud data access with multi-authorities. In 2013 Proceedings IEEE INFOCOM. 2625-2633.

[10] T. Jung, X. Y. Li, Z. Wan, and M. Wan. Control Cloud Data Access Privilege and Anonymity with Fully Anonymous Attribute-Based Encryption. IEEE Transactions on Information Forensics and Security, 2015, 10(1): 190-199.

[11] Allison Lewko, Brent Waters. Decentralizing Attribute-based Encryption. In Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2011). 2011, 568-588.

[12] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-Based Encryption for Circuits from Multilinear Maps. International Cryptology Conference, 479-499.

[13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based Encryption for Circuits. In Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC 2013), 545-554.

[14] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-Based Signatures. CT-RSA 2011, LNCS 6558, pp. 2011,376-392.

[15] Jin Li, Man Ho Allen Au, Willy Susilo, Donggang Xie, Kui Ren. Attribute-based signature and its applications. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010), 2010, 60-69.

[16] Qingji Zheng, Shouhuai Xu, and Giuseppe Ateniese. VABKS: Verifiable attributebased keyword search over outsourced encrypted data. In INFOCOM 2014, 522-530.

[17] S.Guo, Y.Zeng. Attribute-Based Signature Scheme. 2008 International Conference on Information Security and Assurance (ISA 2008), 509-511.

[18] Dalia Khader. Attribute Based Group Signatures. Cryptology ePrint Archive Report 2007/159, 2007.

[19] Camenisch J, Lysyanskaya A. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology. Springer-Verlag, 2001, 93-118.

[20] Zhang J, Xue N, Huang X. A Secure System For Pervasive Social Network-Based Healthcare. IEEE Access, 2016, 4(99): 9239-9250.

[21] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, Andrew Lippman. MedRec: Using Blockchain for Medical Data Access and Permission Managemen-t. International Conference on Open and Big Data, 2016, 25-30.

[22] Ariel Ekblaw, Asaph Azaria, John D. Halamka, Andrew Lippman. A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data. White Paper, August 2016.

[23] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In Proceedings of the 2013 Conference on Internet Measurement Conference (IMC 2013), 127-140.