Practical Medical Files Sharing Scheme Based on Blockchain and Decentralized Attribute-Based Encryption

JIYU TAO1 AND LI LING1,2
1School of Information Science and Technology, Fudan University, Shanghai 200433, China
2Shanghai Blockchain Technology Association, Shanghai 200438, China
Corresponding author: Jiyu Tao (jytao18@fudan.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2101103.

ABSTRACT Medical files can help people prevent diseases, increase cure rates, promote medical development and help solve major public health crises. However, medical files are strongly private. It is an urgent problem needed to be solved that how to share medical files with privacy and data security. The existing models based on the centralized certificate authority is a feasible method, but it is possible to experience a single point failure. Besides there is a mismatch between the models and real-life scenario since they are only suitable for single patient. Therefore, this paper proposes a practical medical file sharing scheme based on blockchain and decentralized attribute-based encryption. The blockchain is used to record application and grant of authorizations. Smart contracts provide an interactive platform for all users in the system. By utilizing decentralized attribute-based encryption, fine-grained access control of medical files is carried out to ensure privacy and security as well as avoiding single point failure. Attribute-based algorithm that support multi-person democratic decision making and dynamic personnel changes are designed to make the model much closer to the real scene. Finally, through security, performance and comparing analysis with other solutions, the scheme in this paper can meet the needs of real-life scenarios in terms of security and practicability, and provides a new practical model for medical file sharing.

INDEX TERMS Attribute-based encryption, blockchain, medical data sharing, access control, authorization.

I. INTRODUCTION
In the traditional medical field, paper medical records and imaging materials have already begun to transform into computer-stored electronic data. Medical files such as electronic medical records, CT, B-ultrasound, and MRI images are mainly stored in the hospital database. Medical files can provide theoretical references for follow-up treatment, so that doctors can propose particular treatment methods based on changes in various indicators. At the same time, timely sharing of suspicious cases can provide data support for major public health crises such as the COVID-19 [1], enabling doctors and scholars around the world to conduct joint research and speed up the progress of the epidemic control.

However, data sharing between different hospitals is still in its initial stage. The electronic medical files of patients cannot be carried out from the hospital. They can only rely on handwritten medical records and printed images. A large number of repeated examinations occur during referral, which increases the financial burden of patients. The lack of information exchange may even cause the patient miss the best treatment time. In addition, the patient’s medical files are extremely private. Once disclosed, it will increase the patient’s psychological burden, cause medical disputes and corresponding social problems. Therefore, there is an urgent need for a medical file sharing scheme that can not only ensure the safety of patient information, but also appropriately share medical files with specific targets.

In order to realize the sharing of medical files and allow patients to control their own medical files to the maximum extent at the same time, some scholars consider using Attribute-based Encryption (ABE) to control access to medical files. ABE has been used in many fields, such as Internet of Things [2] and cloud computing [3], it can reduce communication and computing overhead while sharing information...
in one-to-many scenarios. It can also be used for access control and permission distribution. ABE can be divided into KP-ABE (Key-Policy Attribute-based Encryption) and CP-ABE (Ciphertext-Policy Attribute-based Encryption). Most medical file management scheme uses CP-ABE. Alshehri et al. [4] proposed a scheme which stores data in the cloud and use ABE to control the access list to achieve fine-grained division. Gandikota and Reddy [5] used searchable symmetric encryption and ABE to generate hidden access control structures and keywords based on pseudo-random numbers to ensure data security. Mhatre and Nimkar [6] proposed a multi-authority ABE with a federal structure. Yang et al. [7] introduced a third-party authorization agency to control the distribution and grant of attribute keys, and handed over computing and storage to cloud service providers to reduce the pressure on ordinary users and hospital institutions. Tu et al. [8] adopted fixed-length keys and introduced version number marking and proxy re-encryption to achieve attribute revocation and improved efficiency. However, in these solutions, the data is stored in the data center of the cloud service provider where third-party service providers cannot be supervised. So the integrity and authenticity of the data cannot be guaranteed, which may cause data loss or tampering.

Blockchain is a popular emerging technology in recent years. Its origin can be traced back to 2008. A paper proposed by a researcher under the pseudonym Nakamoto [9] described the blockchain as a non-tamperable, decentralized, open, transparent and traceable ledger. Ethereum [10] and smart contracts [26] further improve the blockchain, allowing the blockchain that was originally used as the ledger to be applied in various industries, especially the medical industry [11]. Blockchain technology provides support for data security. Many scholars have combined the two technologies to propose a new medical file sharing scheme based on blockchain and ABE. Wang et al. [12] proposed an attribute-based blockchain data access control model, designed multiple roles including authorization center, enterprise, etc. and gave the overall process, but lacked specific algorithm implementation. Literature [13]–[16] used ABE and a centralized authority management agency, and stored data on the cloud, and literature [13] used the Inter Planetary File System (IPFS) for distributed storage of files. Literature [17]–[19] adopted a multi-authority scheme. Zhang et al. [17] used cloud-chain collaboration and searchable attribute encryption and supported attribute revocation. Guo et al. [18] put the scene in medical care and used KP-ABE instead of CP-ABE, so that it is convenient for the patients to update their key. Jin et al. [19] realized the revocation of permission by adding a software-defined network and called smart contracts to implement its business logic.

The literatures above all set up a key distribution center to control the permission. These centers generate and distribute the attribute key according to the requirements of the patients. However, there are two shortcomings in these schemes. One is that the centralized authority management agency is prone to single-point failures that will reduce the overall performance of the system. At the same time, there is a risk of inside job within the centralized agency. The second is that the file users and file owners in these papers are based on single-person model. However, in reality, the scientific research institutions, hospital departments and patient associations are all formed by multiple people. The decision to share medical records is the result of democratic discussions among participants. Therefore, a medical file sharing scheme that is closer to reality, supports multi-person negotiation and decentralized access control is needed.

Based on the two shortcomings of the above existing solutions, this paper proposes a practical medical file sharing solution based on blockchain and decentralized attribute encryption. The centralized authority management agency is canceled. We improve the decentralized attribute-based encryption and a multi-person collaboration mechanism is added to support democratic discussions and members to form an association. We take advantage of the construction of attribute keys and design sign and verification mechanism to ensure data integrity. Considering the mobility of personnel, an extensible (t, n) threshold secret sharing algorithm is introduced. We also propose a key update algorithm so that our scheme can finally achieves a flexible, efficient, scalable and more realistic medical file sharing solution.

The rest of this paper is arranged as follows. Section II introduces some preliminaries used in the paper. The system model are proposed in Section III. The detail of our medical files sharing scheme is presented in Section IV. In Section V, we conduct performance, security and comparative analysis. And in Section VI, we give out conclusion and future prospection.

II. PRELIMINARIES
A. BILINEAR MAPPING
Suppose $G_1, G_2$ and $G_T$ are finite multiplicative cyclic groups of prime order $q$ respectively. The bilinear mapping $e : G_1 \times G_2 = G_T$ is established then:

a. Bilinear: $\forall g_1 \in G_1, g_2 \in G_2$ and $\forall x, y \in \mathbb{Z}_q$ satisfy $e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$.

b. Non-degeneration: $\exists g_1 \in G_1, g_2 \in G_2$, such that $e(g_1, g_2) \neq 1_{G_T}$, where $1_{G_T}$ represents the identity element in group $G_T$.

c. Computability: There exists an algorithm $\forall g_1 \in G_1, g_2 \in G_2$ that can get $e(g_1, g_2)$ through calculation.

The elements in the group also satisfy the properties of the multiplicative group, namely $\forall g_1, g_2 \in G_1, g_3 \in G_2$, $e(g_1 g_2, g_3) = e(g_1, g_3) \times e(g_2, g_3)$.

B. EXTENSIBLE (t, n) THRESHOLD SECRET SHARING SCHEME
Shamir [20], Blakley [21] respectively proposed two (t, n) threshold secret sharing schemes. In the scheme, the secret $Y$ to be shared is decomposed into $n$ sub-secrets containing partial information and handed over to $n$ individuals. Only
when \( t \) or more of \( n \) individuals cooperate to solve the secret \( Y \). In this paper, an extensible threshold secret sharing that does not require a trusted third party is applied to meet the needs of decentralized attribute-based encryption and dynamic changes of personnel. The scheme are presented as follows.

Let the secret shared by the association be \( Y = \sum_{i=1}^{n} x_i \), the members of the association \( P_i \) randomly selects a polynomial of \( t - 1 \) order \( f_i(x) = f_{i0} + f_{i1}x + \cdots + f_{i,t-1}x^{t-1} \), where \( f_i(0) = f_{i0} = x_i \). \( P_i \) calculates \( f_j(j) \) \( \forall j \) and sends it to all other members \( P_j, j = 1, 2 \ldots n \) through a trusted channel. After \( P_j \) receives all \( f_j(j)(i \neq j) \), he calculates \( \text{key}_Y = \sum_{i=1}^{n} f_i(j) \) as his part of secret. Let \( X(x) = \sum_{i=1}^{n} f_i(x) \), then the secret to be shared is \( X(0) = \sum_{i=1}^{n} f_i(0) = Y \). \( X(0) \) can be obtained through Lagrange interpolation from any \( t \) of \( n \) parts of shares.

When a new member \( P_v \) participates in, \( P_i \) generates a new polynomial of \( t - 1 \) order \( h_i(x) = h_{i0} + h_{i1}x + \cdots + h_{i,t-1}x^{t-1} \), where \( h_i(0) = 0 \). Then \( P_i \) calculates \( h_i(j) \) and sends it to \( P_j \). \( P_j \) calculates a temporary \( \text{key}_Y, \text{temp} = \sum_{i=1}^{n} h_i(j) \) and sends it to \( P_v \). After \( P_v \) receives any \( t \) of \( n \) \( \text{key}_Y, \text{temp} \), he can obtain \( X(z) \), which is its part of secret through Lagrange interpolation as \( X'(z) = \sum_{i=1}^{n} f_i(x) + \sum_{i=1}^{n} h_i(x) \) and \( h_i(z) = 0 \forall z \).

When a member \( P_u \) leaves, \( P_i \) generates a new polynomial of \( t - 1 \) order \( g_i(x) = g_{i0}x + \cdots + g_{i,t-1}x^{t-1} \), where the constant term is 0. Then \( P_i \) sends \( g_i(j) \) to \( P_j(j \neq i, u) \). \( P_j \) calculates \( \text{key}_Y, \text{new} = \sum_{i=1}^{n} g_i(j) \) as his new part of secret. Since the constant term is 0, it is ensured that the partial secret of leaving member \( P_u \) loses its effect while the secret \( Y \) remains unchanged.

When the threshold value \( t \) needs to be increased to \( t' \), \( P_i \) generates a new polynomial of \( t' - 1 \) order \( k_i(x) = k_{i0}x + \cdots + k_{i,t'-1}x^{t'-1} \). Then \( P_i \) sends \( P_j(k_i(j) \neq i) \). \( P_j \) calculates \( \text{key}_Y, \text{new} = \sum_{i=1}^{n} k_i(j) \) as his new part of secret. Since the constant term has not changed, the secret \( Y \) remains unchanged. Meanwhile, the threshold is raised to \( t' \), which means at least \( t' \) people’s permission is required to reconstruct secret \( Y \).

### C. CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION

Amit Sahai and Brent Waters first proposed the concept of ABE in the paper [22]. The key and ciphertext are respectively marked by several attributes, and each attribute can be defined by the user. In this article, we use the CP-ABE architecture proposed by Lewko and Waters [23] in 2011. In this scheme, all users are authorizing institutions with same authority. There is no administrator or third party involved. This scheme is more suitable for the decentralized property of the blockchain and real-life scenarios.

Lewko’s key policy attribute-based encryption usually consists of the following steps:

a. Global initialization: According to the algorithm, the security parameter \( \lambda \) is input and outputs global parameter \( GP \).

b. Attribute Authority Initialization: Each authority inputs the global parameter \( GP \), and output the public and private key pair of the authority.

c. Data encryption: Encryption algorithm takes in access control policy \( L \), global parameters \( GP \), plaintext \( m \), attribute related public key \( PK \) and outputs ciphertext \( M \).

d. Access Grant: Algorithm takes in the authorized person’s \( UID \), the global parameter \( GP \), a certain attribute \( attr \) of the grantor and the related private key \( SK \) of the attribute and outputs the attribute key \( Key_{UID, attr} \).

e. Data decryption: Decryption algorithm takes in \( Key_{UID, attr} \), global parameters \( GP \), ciphertext \( M \) and outputs either plaintext \( m \) when the key satisfies the attribute requirements of ciphertext or a fail warning.

### III. SYSTEM MODEL

#### A. SYSTEM NOTATION ILLUSTRATION

The notations related to the scheme in this paper are illustrated in Table 1.

| Notation       | Meaning                                      |
|----------------|----------------------------------------------|
| \( GP \)       | the global parameter                         |
| \( UID \)      | the unique identity                          |
| Mem_{pub}, Mem_{pri} | the member’s personal public and private keys |
| Assopub_{i}, Assopri_{i} | the each member’s part of the association public and private keys |
| Assopub, Assopri | the public and private key of the association attributes |
| Attrpub, Attrpri | the attribute public and private keys of different attributes |
| Attr_{pub, i}, Attr_{pri, i} | the each member’s part of the attribute public and private key of different association attributes |
| Attr_{pub}, Attr_{pri} | the attribute public and private key of different association attributes |
| Key_{sym}      | the symmetric encryption key                 |
| CT_{sym}       | the symmetric encryption ciphertext          |
| Sig            | the signature                                |
| Key_{sym, attr} | the attribute’s unique attribute key of attr  |
| CTA_{attr}     | the attribute encryption key                 |
| F              | the medical files                            |
| M              | the access control matrix                    |
| DDB            | the digest data block                        |

#### B. SYSTEM ARCHITECTURE

The overall model of the practical medical file sharing scheme is shown in Figure 1. The entire system can be divided into four kinds of entities. They are member, association, blockchain and DSC.

**Member** (Mem) is the smallest unit in the system. Anybody such as patients and doctors can be member of the system. A member can both be a data owner who symmetrically encrypts and uploads his medical files and a data user who downloads medical files with others’ permission. Each member is equal which means nobody has privilege.

**Association** (Asso) is set in this paper in order to be closer to the real scene. It is composed of multiple members and can grant other person the authority of the association attribute. However, every decision requires joint democratic voting.
Other person will be granted the authority only if a certain number of members agree to authorize.

**Blockchain** and **smart contract** are the basis of the system. Multiple hospitals can join together to form a consortium blockchain. A faster consensus algorithm such as PBFT can be applied to achieve quick transaction instead of power and time-consuming consensus algorithm like PoW in bitcoin. Data blocks of public parameters and digest information are stored on the blockchain after attribute encryption. Smart contracts are deployed to perform operations such as registration and authorization. The blockchain can be led by either the government or several large hospitals to achieve a more secure environment.

**Data Storage Center (DSC)** is responsible for storing the patient’s medical files. Some researchers have proposed that data can be encrypted and stored in the interplanetary file system IPFS, such as paper [7] and paper [24]. In this paper, the data storage is still designed to be symmetrically encrypted and stored in the corresponding database of the hospital which can be extended to IPFS in future work.

### C. SYSTEM OPERATION PROCESS

The operation process of the system is shown in Figure 1 as number sequence. It is worth noting that the communication between members and between members and association is initiated by smart contracts.

**Step ①**: Global initialization. This step is automatically completed when the smart contract is deployed, including initializing system generator, bilinear pair, etc.

**Step ②**: Association registration. After negotiation, members who want to form the association cooperatively interact with the blockchain. They set the total number of members $n$ and the threshold $t$ required to agree the voting. Then generate the public and private keys of the association through the smart contract and upload the public key to the blockchain.

**Step ③**: Attribute registration. Members and Associations initiate attribute registration to the blockchain according to their own needs. Members can directly obtain the attribute public-private key pair. The association needs to cooperatively generate public and private keys through the smart contract. Both member and association need to upload public key to the blockchain.

**Step ④**: Data storage. Any member or association can upload the symmetrically encrypted medical files and store them in the DSC. They then upload attribute encrypted data block which contains the hash of the data, the uploader’s signature, the uploader’s public key and the symmetric encryption key to the blockchain.

**Step ⑤**: Attribute apply and authorization. Members can apply to other members or associations for their registered attribute. Others can decide whether to grant permission or not. If they agree, a unique attribute key will be returned to the applicant. Applying for permission from an association requires the consent of a certain number of people in the association. The applicant collects permission from members so
that he can calculates the unique attribute key and preserves it properly.

**Step 6:** Obtain the data and decrypt. The member try to obtain the attribute ciphertext from the blockchain. If the corresponding attribute requirements are met, the ciphertext can be unlocked to obtain medical file related information in the DSC, and then medical files can be obtained after symmetric decryption.

**IV. A PRACTICAL MEDICAL FILES SHARING SCHEME**

This section will be divided into three sub-sections. The first sub-section introduces the practical characteristics of this scheme which is different from existing schemes. In the second part we will introduce the operation process and cryptographic structure of the scheme proposed in the section III in detail. The third sub-section proposes a key update algorithm considering the mobility of personnel in real-life scenes.

**A. PRACTICAL CHARACTERISTICS OF OUR SCHEME**

In order to solve the problems that may arise from traditional centralized authority granting agencies, this paper proposes a decentralized ABE scheme which uses attribute-based encryption to achieve fine-grained access control. Our scheme has the following characteristics comparing to existing schemes.

1) **SUPPORT MULTI-PERSON DEMOCRATIC VOTING**

Existing medical file sharing schemes are based on single patient model. However, in reality, there are many multi-person scenarios such as hospital departments, scientific research institutions, and concern groups with patients suffering from same disease. Sharing data between institutions requires multiple people to participate in decision-making. The concern group has gathered a large number of cases with the same symptoms but individual differences, which can be a valuable source of scientific research data. Therefore, by introducing threshold secret sharing, this solution realizes that authorization can only be obtained with the consent of a majority of people, which is more suitable for real-life scenarios.

2) **DECENTRALIZATION OF THE AUTHORIZATION PROCESS**

In order to cooperate with the decentralization of blockchain, decentralized attribute-based encryption is utilized in this paper, which avoids the single point failure and internal corruption of centralized CAs. All public-private key pairs are generated by smart contracts. The algorithm is transparent. The right of authorization is handed over to every single member so that each patient has full control of his medical files.

3) **FLEXIBLE ATTRIBUTE REGISTRATION**

Different from the system initialization in some solutions, our scheme can register attributes to the blockchain at any time according to the various need of patients or association. There is no need to pre-define attributes during system initialization which means it costs less time.

4) **SUPPORT DYNAMIC PERSONNEL CHANGE**

Our scheme takes into account the dynamic changes of personnel in the association, including participation and departure of members and increase of the voting threshold. Key update algorithm is designed so that the part held by the left member will automatically become invalid while the public-private key pairs of the association will not change, ensuring the security of the association secret as well as new members’ right.

5) **ENSURE DATA INTEGRITY**

Based on the cryptographic structure, our scheme designs the file signature and verification algorithm to ensure the integrity of the file stored on the blockchain. Any visitor or data owner can check whether the file has been tampered at any time and make corresponding operation promptly.

**B. OPERATION PROCESS AND CRYPTOGRAPHIC STRUCTURE**

- **Global Initialization** $(\lambda) \rightarrow (GP)$: Let $G_1$, $G_2$ and $G_T$ be cyclic multiplicative group of order big prime order $q$. In this paper, we use symmetric group where $G_1$ is the same as $G_2$ and let $g$ be the generator of $G_1$. Let $e : G_1 \times G_2 = G_T$ be a bilinear map. Define a hash function $Q(\text{Hash})$ from $UID$ to $G_1$ group: $2^{256} \rightarrow G_1$ or $G_2$. So, the global parameter is $GP = \{e, G_1, G_T, g, Q(\text{Hash})\}$.

- **Member Initialization** $(GP, UID) \rightarrow (\text{Mem}_{pub}, \text{Mem}_{pri})$: Member registers to the blockchain. He randomly chooses $\gamma \in Z_q$. Then he uploads $\text{Mem}_{pub} = \{e(g, g)^\gamma, g^2\}$ as public key and keeps $\text{Mem}_{pri} = \gamma$ as his private key.

- **Association Registration**

  1) **Association Initialization** $(GP, \{UID_1 \ldots UID_n\}) \rightarrow (\text{Asso}_{pub,j}, \text{Asso}_{pri,j})$: For a group of members $\{\text{Mem}_1, \text{Mem}_2 \ldots \text{Mem}_n\}$ they need to determine the threshold $t$ of passing any decision first. Then $\text{Mem}_i$ needs to randomly choose $\gamma_i \in Z_q$ and $t - 1$ order polynomial $f_i(x) = f_0 + f_1x + \ldots + f_{t-1}x^{t-1}$, where $f_i(0) = f_0 = \gamma_i$, so that the association secret is $\gamma = \sum_{i=1}^{n} \gamma_i$. $\text{Mem}_i$ calculates $f_i(UID_i), j \neq i$ for each other member $\text{Mem}_j$ and sends it to $\text{Mem}_j$. $\text{Mem}_j$ computes $\text{Asso}_{pri,j} = \sum_{i=1}^{n} f_i(UID_i)$ and $\text{Asso}_{pub,j} = e(g, g)^{\text{Asso}_{pri,j}}$, $\text{Asso}_{pub,j} = g^{\text{Asso}_{pri,j}}$. Then uploads $\text{Asso}_{pub,j} = \{\text{Asso}_{pub,a}, \text{Asso}_{pub,b}, \text{Asso}_{pub,f}\}$ to the blockchain.

  2) **Association Key Assemble** $(\text{Asso}_{pub,j})^{(j)} \rightarrow (\text{Asso}_{pub})$: All members submit their part of association public keys to the smart contract so that the association public key $\text{Asso}_{pub} = \{\text{Asso}_{pub,a}, \text{Asso}_{pub,b}, \text{Asso}_{pub,f}\}$ can be calculated by the smart contract:

  $$\text{Asso}_{pub} = e(g, g)^{\sum_{i=1}^{n} \gamma_i} = e(g, g)^{\sum_{j=1}^{n} (\text{Asso}_{pri,j} \prod_{i=1, i\neq j}^{n} UID_i, UID_j)}$$
Then the smart contract automatically uploads public key Asso\textsubscript{pub} to the blockchain.

- **Attribute Registration**
  1) Member Attribute Registration\((GP, UID) \rightarrow (Attr\textsubscript{pub}, Attr\textsubscript{pri})\): Member registers a unique attribute \textit{attr} to the blockchain. He randomly chooses \(\tau_{attr} \in \mathbb{Z}_q\). Then he uploads \(Attr\textsubscript{pub} = g^{\tau_{attr}}\) as public attribute key and keeps \(Attr\textsubscript{pri} = \tau_{attr}\) as his private attribute key.
  2) Association Attribute Initialization\((GP, \{UID_1 \ldots UID_n\}) \rightarrow (Attr\textsubscript{pub}, Attr\textsubscript{pri})\): Similar to the attribute initialization, \(Mem_i\) randomly selects \(\tau_i \in \mathbb{Z}_q\) and a \(t-1\) order polynomial \(h_i(x) = h_0 + h_1x + \ldots + h_{t-1}x^{t-1}\), where \(h_i(0) = h_0 = \tau_i\). Then calculates \(h_i(UID_j), j \neq i\) for each other member \(Mem_j\) and sends it them. \(Mem_j\) computes \(Attr\textsubscript{pri,j} = \sum_{i=1}^{d} h_i(UID_i)\) and \(Attr\textsubscript{pub,j} = g^{Attr\textsubscript{pri,j}}\) and uploads \(Attr\textsubscript{pub,j}\) to the blockchain.
  3) Association Attribute Key Assemble\((Attr\textsubscript{pub,j} \forall j) \rightarrow (Attr\textsubscript{pub})\): All members submit their part of association attribute public keys to the smart contract so that \(Attr\textsubscript{pub}\) can be calculated by the smart contract:

\[
\begin{align*}
Attr\textsubscript{pub} &= g^{\sum_{j=1}^{\delta} \tau_i} \\
&= g^{\sum_{j=1}^{\delta} (Attr\textsubscript{pri,j} \prod_{i=1}^{\delta} UID_i^{-1})} \\
&= \prod_{j=1}^{\delta} Attr\textsubscript{pub,j}^{\prod_{i=1}^{\delta} UID_i^{-1}}
\end{align*}
\]

Then the smart contract automatically uploads \(Attr\textsubscript{pub}\) to the blockchain.

- **Data Encryption**
  1) Medical Files Encryption\((F, Key\textsubscript{sym}) \rightarrow (CT\textsubscript{sym}, Q(CT\textsubscript{sym}), Sig)\): The uploader chooses a symmetric key \(Key\textsubscript{sym}\) and a symmetric encryption function \(E\) and computes \(CT\textsubscript{sym} = E(F, Key\textsubscript{sym})\). He then calculates \(Q(CT\textsubscript{sym})\) with the hash function and uses his own private key \(Mem\textsubscript{pri} = \gamma\) to sign \(Sig = e(Q(CT\textsubscript{sym}), g)^\gamma\). The signature algorithm has been proved safe in paper [25]. Finally, the data block \(\{CT\textsubscript{sym}, Q(CT\textsubscript{sym}), Sig\}\) is uploaded and preserved in DSC.

The association uploading a medical file together requires each member to use his \(Asso\textsubscript{pri,i}\) to sign \(Sig_j = e(Q(CT\textsubscript{sym}), g)^{Asso\textsubscript{pri,i}}\), then upload the data block\(\{CT\textsubscript{sym}, Q(CT\textsubscript{sym}), [Sig_j, \forall j]\}\) to the DSC to preserve.

2) Digest Data Encryption\((DDB, Mem\textsubscript{pub}, Asso\textsubscript{pub}, Attr\textsubscript{pub}, Attr\textsubscript{attr}) \rightarrow (CT\textsubscript{attr})\): The ciphertext contains four parts \(CT\textsubscript{attr} = \{Part_0, \{Part_{1,i}, Part_{2,i}, Part_{3,i} \forall i\}\). The structure of each part is given as follows:

\[
\begin{align*}
Part_0 &= DDB \cdot e(g, g)^s \\
Part_{1,i} &= e(g, g)^{M_{1,v}} \cdot e(g, g)^{v \cdot \gamma_i} \\
Part_{2,i} &= g^{\tau_i} \cdot Part_{3,i} = g^{\tau_i} \cdot g^{M_{1,w}}
\end{align*}
\]

where \(Part_0, Part_{1,i} \in G_T, Part_{2,i} \in G_1\).

For \(m\) attributes in the access control matrix, \(Part_{1,i}, Part_{2,i}, Part_{3,i}\) are arrays of length \(m\), which store the elements of each attribute separately. \(DDB\) is plaintext and \(s\) is randomly selected from \(Z_q\). \(M_i\) is the \(i\)th row of the access control matrix which represents the access control policy of the attribute \(i\). \(v\) is a column vector of length \(d\), in which the first element is \(s\). \(e(g, g)^{\gamma_{1,v}}\) is the member or association public key related to attribute \(i\) \(g^{\gamma_i}\) is the member or association attribute public key related to attribute \(i\). \(\gamma_i\) is randomly selected from \(Z_q\). \(w\) is a column vector of length \(d\), in which the first element is zero and the rest are randomly selected. Finally, \(CT\textsubscript{attr}\) is uploaded to the blockchain after encryption.

- **Attribute apply and authorization**
  1) Member Authorization\((UID, Mem\textsubscript{pri}, Attr\textsubscript{pri}) \rightarrow (Key_{a,attr})\): \(Mem_a\) tries to apply for the permission of attribute \(attr\) from \(Mem_{ja}\). \(Mem_{ja}\) calculates \(Key_{a,attr} = g^{\gamma_i Q(UID_j)^{\gamma_{1,v}}}\) if he agrees and sends it to \(Mem_a\). \(Mem_a\) preserves it properly for future decryption.
  2) Association Authorization\((UID, Mem\textsubscript{pub}, Attr\textsubscript{pub,\forall j}) \rightarrow (Key_{a,attr})\): \(Mem_a\) tries to apply for the attribute \(aattr\) from \(Asso\). \(n\) association members \\{\(Mem_1 \ldots Mem_n\)\} will receive the request and decide whether to authorize \(Mem_a\). If \(Mem_{ja} \in n\) agrees to the authorize, then calculates \(Key_{a,attr} = g^{\gamma_{1,v} Q(UID_j) \gamma_i}\) and transmits it to \(Mem_a\).

If \(Mem_a\) receives the authorization of more than \(m\) members of \(Asso\), he can calculate:

\[
Key_{a,attr} = \prod_{j=1}^{\delta} (Asso_{pri,j} \prod_{i=1}^{\delta}UID_i^{-1})
\]

\[
\begin{align*}
&= g^{\sum_{j=1}^{\delta} (Asso_{pri,j} \prod_{i=1}^{\delta}UID_i^{-1})} \\
&\cdot Q(UID_j) \prod_{j=1}^{\delta} (Asso_{pri,j} \prod_{i=1}^{\delta}UID_i^{-1})
\end{align*}
\]

\[
\prod_{j=1}^{\delta} Key_{a,attr} = Key_{a,attr} = g^{\gamma_{1,v} Q(UID_j)\gamma_i}
\]

\(Mem_a\) preserves it properly for future decryption.

- **Data Decryption**
  1) Digest Data Decryption\((CT\textsubscript{attr}, Key_{a,attr}) \rightarrow (DDB)\): \(Mem_a\) needs to use his unique key \(Key_{a,attr}\) to decrypt medical files. He needs to perform the following operations on each attribute \(i\) in the access control matrix:

\[
\begin{align*}
Part_{1,i} &= e(Q(UID_j), Part_{3,i}) \\
&= e(g, g)^{M_{1,v}} \cdot e(Q(UID_j), g^{M_{1,w}})
\end{align*}
\]
For each row $M_i$ of $M_{m \times d}$, find a $\beta_i$ such that 
$$\sum_{m=1}^{m} \beta_j M_i = [1, 0, \ldots, 0]$$
and then multiply results together:
$$\prod_{i=1}^{m}(e(g, g)^{M_i}, e(Q(UID_a), g^{M_i}))^{\beta_i} = e(g, g)^{t}$$
where $\sum_{m=1}^{m} \beta_j M_i \cdot v = s$ and $\sum_{m=1}^{m} \beta_j M_i \cdot w = 1$. Then we can get $DDB = Part0(e(g, g)^{t})$.

2) **Medical Files Decryption** ($DDB$) $\rightarrow$ ($F$, bool): $Mem_a$ can find ($CT_{sym}$, $H(CT_{sym})$, $Sig$) in $DSC$. He first compares the hash and signature decrypted from $DDB$ with those stored in $DSC$. If they are different, then $bool = False$. Otherwise $Mem_a$ calculates $Q(CT_{sym})$ according to the stored $CT_{sym}$, if it differs from the one stored in $DSC$, $bool = False$. Finally, $Mem_a$ computes $e(Q(CT_{sym}), g^{y})$ with the public key $g^{y}$ stored in $DDB$ and verifies whether it is the same as $Sig = e(H(CT_{sym}), g^{y})$. If same, then $bool = True$, the medical file is intact. He can use the decryption function $D$ and the symmetric key $Key_{sym}$ in $DDB$ to decrypt $CT_{sym}$ and obtain medical file $F = D(CT_{sym}, Key_{sym})$.

For a medical file uploaded by an association, $Mem_a$ first assembles a set of signatures {$Sig_i$, $y_i$}:
$$Sig = e(H(CT_{sym}), g^{y}) = e(H(CT_{sym}), g^{\sum_{i=1}^{n} y_i}) = e(H(CT_{sym}), g^{\sum_{i=1}^{n} Asso_{pubi}^{\gamma_i}})^{\prod_{i=1}^{n} \frac{UID_i^{\gamma_i}}{BD_i^{\gamma_i}}} = \prod_{i=1}^{n} \frac{UID_i^{\gamma_i}}{BD_i^{\gamma_i}}^\gamma_i$$

He then computes $e(H(CT_{sym}), Asso_{pubi}) = e(H(CT_{sym}), g^{y})$ and verifies whether it is the same as $Sig$. If same, then $bool = True$ and the medical file is intact.

### C. **KEY UPDATE ALGORITHM**

In real-life scenarios, the members within an association are not static. In order to ensure that part of the secrets held by the leaving members become invalid and to prevent the original threshold from being too low when there are too many members, our scheme sets up a key update algorithm.

1) **PARTICIPATION OF NEW MEMBERS**

When a new member $P_z$ joins, $P_z$ generates a new polynomial of $t - 1$ order $h_t(x) = h_0 + h_1x + \ldots + h_{t-1}x^{t-1}$, where $h_t(UID_z) = 0$. Then $P_z$ calculates $h_t(UID_j)$ and sends it to $P_i$, $P_j$ makes up his discussion on whether he agrees $P_z$ to become new member, if he agrees, he calculates temporary $key_{j, temp} = Asso_{pri}, Asso_{pubi}^{\gamma_i} + \sum_{i=1}^{n} h_i(UID_j)$ and $key_{i, temp, attr} = Asso_{pubi}^{\gamma_i} + \sum_{i=1}^{n} h_i(UID_j)$ and sends them to $P_z$. If $P_z$ receives any $t$ of $n$ permissions, he can obtain his part of association private key through Lagrange interpolation as $X'(UID_z) = X'(UID_z) + \sum_{i=1}^{n} h_i(UID_z)$ and $h_i(UID_z) = 0 \forall z$:

$$Asso_{pri, x} = X(UID_z) = \prod_{i=1}^{t} \frac{UID_i^{\gamma_i} - UID_z^{\gamma_i}}{BD_i^{\gamma_i} - BD_z^{\gamma_i}}$$

Then $P_z$ calculates $Asso_{puba, x} = e(g, g)^{Asso_{pri, x}}$, $Asso_{pubb, x} = g^{Asso_{pri, x}}$ and $Asso_{pubp, x} = g^{Asso_{pri, x}}$. Now $P_z$ has his part of association public key and attribute public key while the association public key remains unchanged.

2) **DEPARTURE OF MEMBERS**

When a member $P_i$ exists, $P_i$ generates a new polynomial of $t - 1$ order $g_i(x) = g_{i1}x + \ldots + g_{i, t-1}x^{t-1}$, where the constant term is 0. Then $P_i$ sends $g_i(UID_i)$ to $P_j(j \neq i, u)$. $P_j$ calculates $Asso_{pri}, Asso_{puba, j} = Asso_{pri} + \sum_{i=1}^{n} g_i(UID_i)$, $Asso_{puba, j} = e(g, g)^{Asso_{pri}}, Asso_{pubb, j} = g^{Asso_{pri}}$ as his new part of association keys and $Asso_{pri}, Asso_{puba, j} = Asso_{pri} + \sum_{i=1}^{n} g_i(UID_i)$, $Asso_{puba, j} = g^{Asso_{pri}}$ for each attribute as new association attribute keys. Since the constant term is 0, it is ensured that the part of public or private keys of leaving member $P_i$ become invalid while the association public key remains unchanged.

3) **INCREASE OF THRESHOLD**

When the threshold $t$ needs to be increased to $t'$, $P_i$ generates a new polynomial of $t' - 1$ order $h_{t'}(x) = h_{t'1}x^{t'} + \ldots + h_{t', t-1}x^{t-1}$, then $P_i$ sends $h_{t'}(UID_i)$ to $P_j(j \neq i)$. $P_j$ calculates $Asso_{pri}, Asso_{puba, j} = Asso_{pri}^{\prime} + \sum_{i=1}^{n} h_{t'}(UID_i)$, $Asso_{puba, j} = e(g, g)^{Asso_{pri}^{\prime}}, Asso_{pubb, j} = g^{Asso_{pri}^{\prime}}$ as his new part of association keys and $Asso_{pri}, Asso_{puba, j} = Asso_{pri}^{\prime} + \sum_{i=1}^{n} h_{t'}(UID_i)$, $Asso_{puba, j} = g^{Asso_{pri}^{\prime}}$ for each attribute as new association attribute keys. Meanwhile, the threshold is raised to $t'$, which means at least $t'$ peoples' permission is required to pass a vote.

### V. **MODEL ANALYSIS**

#### A. **SECURITY ANALYSIS**

The cryptographic security of the scheme proposed in this paper has been proven in the literature [23]. This section will analyze several possible attacks on this scheme.

1) **CONSPIRACY ATTACK**

A conspiracy attack refers to multiple members with different attributes conspiring to crack an encrypted file that they cannot decrypt individually. Suppose $Mem_a$ has the authority of $attr_1$ and $Mem_b$ has the authority of $attr_2$. The encryption policy is ($attr_1 AND attr_2$). It must be guaranteed the file cannot be cracked if they cooperate together. In our scheme, the corresponding authority attribute is specific calculated with the $UID$ of the applicant. Even if the conspirators have $g^{\gamma_i}Q(UID_a)^{\tau_{attr_1}}$ and $g^{\gamma_i}Q(UID_a)^{\tau_{attr_2}}$, it is impossible to inversely compute private key $\tau$ and the attribute private key $\tau_{attr_1}$ and $\tau_{attr_2}$ within the polynomial time according to discrete logarithm problem. Therefore, they cannot complete the attack.

For a file encrypted with association attribute, it is more difficult for the attacker due to the existence of the threshold.
2) MAN-IN-THE-MIDDLE ATTACK
For each unique attribute registered by a member, the system will return a one-time random number as the attribute private key, so the key cannot be obtained by a repeated registration. The authorizer will verify the corresponding public key uploaded to the blockchain of the applicant while reviewing the apply and use the public key to encrypt and transmit $\text{Key}_{a, \text{attr}}$ so that only the applicant can decrypt the key.

3) DoS ATTACK
DoS attack refers to the blocking of access requests of ordinary users by establishing a large number of illegal requests at the same time. In a traditional centralized scheme, the single-point centralized authorization agency is the weak point of the entire system. Attackers can launch attacks on the agency and block normal requests. In this paper, our decentralized solution solves the problem. The robustness of the blockchain guarantees the system functioning normally from DoS attack as long as the attacker cannot compromise most of the nodes in the blockchain.

4) DATA INTEGRITY
In this paper, DSC is assumed to be semi-trusted, it will be curious about the data stored in it. Although DSC can freely obtain the data in it, DSC has no advantage over any other attacker as medical files are encrypted with a symmetric key and stored in a ciphertext form. At the same time, the hash of the file has been fixed through the blockchain. Due to the immutability of the blockchain, both the data owner and data user can inspect the files to prevent DSC from tampering the data.

B. PERFORMANCE ANALYSIS
We use a PBFT-based blockchain as the basis of the smart contract. The attribute encryption program is written in Python. The simulation is based on the virtual machine of Ubuntu 18.04LTS. The virtual machine uses an AMD Ryzen 5800X (Zen3) @3.8GHz, 2 Cores 2 Threads and 4g DDR4 memory.

We use AES (advanced encryption standard) as the symmetric encryption function with a 256-bit key as the symmetric encryption is only used for digest data block of which typical data size is several kilobytes. At the cost of 10% more time which is tens of milliseconds, it is much more secure compared to 128 or 192 bits key.

The computational time cost of our scheme is mainly that of calculation in $\mathbb{Z}_q$, $G_1$ and $G_T$. The notations in analysis are as follows:

- $P$: time complexity of a bilinear pairing operation
- $E_q$: time complexity of exponential operation on $\mathbb{Z}_q$
- $E_{G_1}$: time complexity of exponential operation in $G_1$ group
- $E_{G_T}$: time complexity of exponential operation in $G_T$ group
- $a$: number of global attributes
- $m$: number of attributes actually used for encryption
- $u$: number of attributes actually used for decryption
- $t$, $n$: parameters in the threshold secret sharing algorithm

The time of multiplication in $G_1$ or $G_T$ group is almost negligible compared to the time complexity above. At the same time, it ignores hash operation, signature, call of smart contract, network and other overheads. In order to reduce the fluctuation caused by randomness, the simulation runs 5 rounds, 1000 tests per round. Table 2 shows the time complexity of each operation as a benchmark for following analysis.

| Operation | $P$ | $E_q$ | $E_{G_1}$ | $E_{G_T}$ |
|-----------|-----|-------|-----------|-----------|
| Time cost/ms | 8.98 | 0.131 | 7.108 | 0.795 |

Table 3 shows the comparative analysis of the theoretical value of our scheme with Paper [14] and Paper [18] which also utilize blockchain-based ABE scheme. Paper [14] uses the attribute encryption of a centralized authorization agency, therefore the calculation process is simpler. But the centralized solution has some drawbacks and does not support multi-person association. In contrast, paper [18] supports the decentralized model of multiple authorization agencies. There are more steps and all the attributes in this scheme are registered at the beginning of initialization. Each step of authorization, encryption, and decryption contains all the attributes so it is more complicated. Besides, its model is also based on a single patient. In this paper, we use a different encryption method, and the cryptographic structure is different from other schemes. We have a leading position in member registration, member attribute registration, encryption and decryption. Our scheme also supports dynamic changes of personnel. The frequency of association-related operations is lower. It is not necessary to preset all the attributes during the initialization. Members can register at any time on demand. The last column of Table 3 shows the actual time of each step in the simulation. The threshold parameters $n = 3$, $t = 2$, $T = 3$. The number of attributes used for encryption $m = 2$ and the number of attributes used for decryption $u = 2$. As presented, the real time complexity is basically the same as the theoretical result, which can not only support the theoretical result, but meet the demand of real-life scenario as well.

Considering that association attribute registration is the most frequently used and time-consuming step in actual scenarios, Table 4 shows the time it takes to register and generate an association attribute public key. Consider the most severe case, that is, the time complexity of a member and smart contract when all members’ permission is needed to pass a vote and threshold $t = n$. As shown in the table, it does not take the member much time to calculate $h_i(UID)$. During programming implementation, multiplication and addition are used to take place of exponential operations, so the time complexity is proportional to the square of Mem number, which is consistent to the theoretic value in Table 3. Since that time complexity of exponential operation on $\mathbb{Z}_q$ is rather low, it only takes about 500 ms in an association of 500 members.
TABLE 3. Time cost comparison of various schemes.

| Step                  | Paper[14]                  | Paper[18]                  | Our scheme                  | Time /ms |
|-----------------------|----------------------------|----------------------------|----------------------------|----------|
| Global Init           | $P + 2E_1 + E_T$           | $P$                        | $P$                        | 10.261   |
| Mem Registration      | $3E_1$                     | $E_T$                      | $E_1 + E_T$                | 8.829    |
| Asso Registration     | $n \cdot E_T$              | $E_T$                      | $n(t - 2)E_1 + E_T$        | 8.871    |
| Mem Attributes Registration | $3E_1$               | $E_1$                      | SC $t \cdot (E_1 + E_T)$   | 17.498   |
| Asso Attributes Registration | $k \cdot E_T$          | $E_1$                      | Mem: $n(t-2)E_1 + E_T$     | 7.672    |
| Mem Authorization     | $(2a + n - 1)E_1$          | $E_1$                      | Mem: $n(t-2)E_1 + E_T$     | 7.345    |
| Asso Authorization    | $E_T$                      | $E_1$                      | SC $t \cdot E_1$           | 14.504   |
| Attributes Encryption | $2E_T + (3m + 5)E_1$       | $E_T + 2n(n-1)E_1$         | $E_T + n(2E_T + 3E_1)$     | 51.277   |
| Sign                  | $6E_1$                     | $E_T$                      | $P + E_T$                  | 10.301   |
| Attributes Decryption | $4P + u(2P + E_T)$         | $n \cdot u(4P + E_1)$      | $u(2P + E_T)$              | 35.121   |
| Verify                | $4P$                       | $E_T$                      | $P$                        | 9.402    |
| Key Update            | Revocation: $2a \cdot E_1$ | Participation:             |                           | 8.630    |
|                       |                            | New Mem: $n(t-2)E_1 + E_T$ |                           | 16.984   |
|                       |                            | Departure: $n(t-2)E_1 + E_T$ |                           | 8.724    |
|                       |                            | Increase Threshold: $n(t-2)E_1 + E_T$ |                           | 9.156    |

TABLE 4. Time complexity of attributes registration with different n/ms.

| Mem Number n | Mem Calculate $h_i(UID_j)$ | Mem Calculate $AAAttr_{pub,j}$ | SC Assemble $AAAttr_{pub}$ |
|--------------|----------------------------|-------------------------------|--------------------------|
| 5            | 0.0658                     | 7.570                         | 37.195                   |
| 20           | 0.864                      | 7.148                         | 146.239                  |
| 50           | 5.493                      | 7.207                         | 386.938                  |
| 100          | 23.534                     | 7.229                         | 812.941                  |
| 250          | 135.759                    | 7.375                         | 2362.76                  |
| 500          | 553.214                    | 7.625                         | 5846.045                 |

TABLE 5. Encryption and decryption time with different number of attributes/ms.

| Attributes Number | Encryption     | Decryption    |
|-------------------|----------------|---------------|
| 2                 | 48.603         | 43.145        |
| 4                 | 91.315         | 86.849        |
| 8                 | 187.603        | 158.430       |
| 16                | 370.184        | 313.135       |
| 32                | 763.913        | 717.001       |
| 64                | 1515.748       | 1435.595      |

The time cost of member calculating $AAAttr_{pub,j}$ is nearly the same, which is an exponential operation in $G_1$ group. The time complexity of smart contract assembling $AAAttr_{pub}$ is proportional to the number of members. Even in a large group of 500 people, it still remains at the second level. At the same time, it should be noticed that this part of the calculation is completed by the blockchain node, which can make use of the powerful computational power of the node to further reduce time consumption. Also, Table 4 considers the most severe condition. In most circumstances, decisions can be passed with the consent of some people in an association.

Table 5 shows the time complexity of encryption and decryption with different number of attributes. We still consider the most severe case, that is, all attributes need to be satisfied and different attributes are connected with $\text{AND}$. It can be seen from the table that in the case of all $\text{AND}$, the time complexity of encryption and decryption is proportional to the number of attributes. When 64 attributes are involved, that is, 64 conditions need to be met at the same time, the encryption and decryption time still only takes about 1.5 seconds, indicating that our scheme is practical in actual scenarios.

Considering that after the association is founded, new members may participate and old members may leave. Table 6 shows the time complexity of calculating new keys with different thresholds. The time complexity of increasing threshold means increase $t$ on. As shown in the table, the time complexity of member participation, member departure and increasing threshold are all at millisecond level, which won’t cause heavy computational burden to the members. Comparing row 2, 3 and 4 of the table, it can be seen that the time cost of new member when he participates in the association is only related to $t$. When $t$ is fixed, time complexity of former member is proportional to $n$ and the fitting equation is $y = 8.47053 + 0.03988n$. The constant term 8.470 is exactly the time consumption of $E_1 + E_T$ and matches the theoretical analysis in Table 3. Comparing row 6, 7, and 8, when $n$ is fixed, time complexity of former member is proportional to $t$ and the fitting equation is $y = 8.39667 + 0.4139t$. The constant item 8.397 also conforms to the theoretical analysis in Table 3. Comparing row 4, 5, and 8, when the increase of the threshold is the same, the time consumption each member is proportional to $n$ and the fitting equation is $y = 8.57423 + 0.10045n$. The constant term 8.574 is also the time complexity of $E_1 + E_T$. Compared to the time cost of calculating association public key, operation in $G_1$ field is not unnecessary. So the time complexity is shorter.
Besides, $M_{e_j}$ can apply $\sum_{i=1}^{t} h_i(UID_j)$ to all new keys and attributes instead of generating a new polynomial for each attribute, which greatly reduce time consumption when members change.

### C. COMPARING ANALYSIS

In this part, we will compare the scheme proposed in this article with the existing medical file management scheme in terms of function (Table 7). At the same time, we will give the solution to the various problems in a medical file sharing scheme (Table 8).

It can be seen from Table 7 that most of the existing solutions set a centralized CA, which is likely to experience a single point failure. Some early schemes didn’t combine blockchain because they were limited by times. In some solutions, the blockchain is only used to preserve data from being tampered. These schemes ignore the functions that can be realized by smart contracts which is an enormous progress in blockchain technology. One important point is that our scheme supports multi-person. Patients can cooperate together as an association while in other schemes patients are irrelevant individuals and cannot make any contribute to medical development. At the same time, our key update algorithm is used to make sure that association key will not leak. While in other solutions, key update is controlled by CA which is also responsible for attribute revocation. Attribute revocation requires change in either ciphertext or attribute key. Our scheme is design for a decentralized environment, key authorization is conducted by members. Also, changing ciphertext means complete replacement of medical files.

### TABLE 6. Dynamic member change time cost with different $(t, n)/\text{ms}$.

| $(t, n)$ | Add(Former Mem) | Add(New Mem) | Leave | Increase Threshold |
|---------|-----------------|--------------|-------|--------------------|
| (2, 3)  | 8.487           | 16.584       | 8.527 | 8.310              |
| (20, 30)| 9.701           | 165.794      | 9.689 | 9.359              |
| (20, 40)| 10.021          | 166.038      | 10.063| 10.099             |
| (20, 70)| 11.274          | 166.146      | 11.239| 15.757             |
| (50, 100)| 18.998         | 435.642      | 18.625| 18.423             |
| (50, 200)| 28.995          | 435.454      | 28.879| 69.447             |
| (100, 200)| 49.980          | 906.604      | 49.733| 49.670             |
| (150, 200)| 70.385          | 1425.259     | 70.684| 28.710             |

### TABLE 7. Function comparison of various schemes.

| Functions                | Paper [4] | Paper [8] | Paper [14] | Paper [17] | Paper [18] | Our Scheme |
|--------------------------|-----------|-----------|------------|------------|------------|------------|
| Based on Blockchain      | No        | No        | Yes        | Yes        | Yes        | Yes        |
| Fine-grained Access Control | Yes      | Yes       | Yes        | Yes        | Yes        | Yes        |
| Support Key Update       | No        | Yes       | Yes        | Yes        | Yes        | Yes        |
| Data Integrity           | Yes       | No        | No         | Yes        | Yes        | Yes        |
| Support Attribute Revocation | No       | Yes       | No         | Yes        | No         | No         |
| Support Multi-person     | No        | No        | No         | No         | Yes        | Yes        |
| No need of CA            | Yes       | No        | No         | No         | Yes        | Yes        |
| Use Smart Contract       | No        | No        | Yes        | No         | No         | Yes        |

### TABLE 8. Problems of medical files sharing schemes and our solution.

| Category          | Problem                                                      | Our solution                                                                 |
|-------------------|--------------------------------------------------------------|------------------------------------------------------------------------------|
| Privacy security  | $DSC$ may be interest in data                                  | We use symmetric encryption to encrypt the medical files and store them in the $DSC$ to ensure that $DSC$ cannot access to the original file. ABE is used for fine-grained access control of visitors. All authorization and application records are stored on the blockchain, which is transparent and cannot be tampered. |
|                   | Data access record is unknown                                  |                                                                              |
|                   | Sharing history may be tampered                                |                                                                              |
|                   | Abuse of authority, difficult to trace                         |                                                                              |
| Patient participation | Have no control over medical files                             | The patient is the only member who has complete control over the medical file. He can cooperate with other wardmates, share their data with medical research institutions and contribute to the development of medicine. |
|                   | Cannot contribute to medical development                       |                                                                              |
| Data mobility     | Medical files are not shared among hospitals                   | Hospitals form consortium blockchain. Share data through patients’ authorization. Data can be shared among hospitals. |
|                   | Sharing Plaintext may infringe privacy                          |                                                                              |
| Data Integrity    | Data is easy to lose                                          | The medical files are signed by the patient. Hash is calculated. Digest data is stored in the blockchain to ensure data integrity. |
|                   | Authenticity cannot be guaranteed                              |                                                                              |
| Model Practicality | Single point failure and internal corruption                 | This paper proposes a decentralized medical file sharing solution to avoid single point failure and internal corruption of authorization agency. The cryptographic structure is modified to adapt multi-person association. Extensive threshold secret sharing is combined to achieve multi-person voting decision. Key update algorithm is designed to adapt dynamic personnel change. |
|                   | Multi-authority model does not support democratic voting and dynamic change |                                                                              |
|                   | Single patient model does not fit real life scenario           |                                                                              |
files which is time consuming. Therefore, our scheme does not support attribute revocation. However, we can achieve global backward revocation by changing the encryption key of subsequent files.

VI. CONCLUSION AND FUTURE PROSPECTION

In the future, the informatization in medical field will continue to advance. The sharing of medical files is a key point. It can help patients avoid repeated examinations and provide effective data for scientific research. This paper proposes a practical medical file sharing scheme based on blockchain and decentralized ABE. In our scheme, patients have complete control over their medical files and there is no need to worry about privacy leakage. The association mechanism of multi-person cooperation, democratic decision-making and support for dynamic changes of personnel makes the model closer to real-life scenarios. Our decentralized model not only matches the idea of decentralization of the blockchain, but also avoids the management problems caused by centralized institutions. Finally, safety analysis, performance analysis and comparison with other schemes prove that the scheme proposed in this paper is quite feasible and have advantage over existing schemes. In future work, we consider introducing proxy re-encryption and zero-knowledge proof to further simplify the management of patients’ medical files and improve privacy protection.

REFERENCES

[1] L. Zhang, B. Li, P. Jia, J. Pu, B. Bai, Y. Li, P. Zhu, L. Li, G. Zeng, X. Zhao, S. Dong, M. Liu, and N. Zhang, “An analysis of global research on SARS-CoV-2,” J. Biomed. Eng., vol. 37, no. 2, pp. 236–249, Apr. 2020, doi: 10.7507/1001-5515.20200234.

[2] S.-Y. Tan, K.-W. Yeow, and S. H. Hwang, “Design and implementation of electronic medical record storage system based on blockchain,” in Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON), Oct. 2019, pp. 190–196, doi: 10.1109/IEMCON.2019.8936186.

[3] J. Tao, L. Ling: Practical Medical Files Sharing Scheme

[4] L. Zhang, Z. Zheng, and Y. Yuan, “A controllable sharing model for electronic health records based on blockchain,” Acta Automat. Sinica

[5] S. Dong, M. Liu, and N. Zhang, “An analysis of global research on SARS-CoV-2,” J. Biomed. Eng., vol. 37, no. 2, pp. 236–249, Apr. 2020, doi: 10.7507/1001-5515.20200234.

[6] S.-Y. Tan, K.-W. Yeow, and S. O. Hwang, “Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things,” IEEE Internet Things J., vol. 6, no. 4, pp. 6384–6395, Aug. 2019, doi: 10.1109/JIOT.2019.2900631.

[7] H. Wang, D. He, J. Shen, Z. Zheng, C. Zhao, and M. Zhao, “Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing,” Soft Comput., vol. 21, no. 24, pp. 7325–7335, Jul. 2016, doi: 10.1007/s00500-016-2271-2.

[8] S. Alshehri, S. P. Radiszowski, and R. K. Raj, “Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption,” in Proc. IEEE 28th Int. Conf. Data Eng. Workshops (ICDEW), Apr. 2012, pp. 143–146.

[9] G. Ramu and B. E. Reddy, “Secure architecture to manage EHR’s in cloud using SSE and ABE,” Health Technol., vol. 5, nos. 3–4, pp. 195–205, Oct. 2015.

[10] S. Mhatre and A. V. Nimkar, “Secure cloud-based federation for EHR using multi-authority ABE,” in Progress in Advanced Computing and Intelligent Engineering, vol. 714. Berlin, Germany: Springer, 2019, pp. 3–15, doi: 10.1007/978-3-030-224-4_1.

[11] J. Yang, T. Li, X. Pei, L. Wen, and C. Wang, “Medical data sharing scheme based on attribute cryptosystem and blockchain technology,” IEEE Access, vol. 8, pp. 45468–45476, 2020, doi: 10.1109/ACCESS.2020.2976894.

[12] Y. Yu, F. Xia, and G. Yang, “Personal health record using attribute-based access control supporting attribute revocation,” J. Chin. Comput. Syst., vol. 38, no. 4, pp. 834–838, Apr. 2017.

[13] S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[14] G. Wood. (2014). Ethereum: A Secure Decentralised Generalised Transaction Ledger. [Online]. Available: http://gavwood.com/paper.pdf

[15] W. She, J. Chen, Q. Liu, Y. Hu, Z. Gu, Z. Tian, and W. Liu, “New blockchain technology for medical big data security sharing,” J. Chin. Comput. Syst., vol. 40, no. 7, pp. 1449–1454, Jul. 2019.

[16] X. Wang, X. Jiang, and Y. Li, “Model for data access control and sharing based on blockchain,” J. Softw., vol. 6, pp. 1661–1669, Oct. 2019.

[17] Y. Xiao, “Design and implementation of electronic medical record storage system based on blockchain,” M.S. thesis, School Comput. Sci. Technol., Dalian Univ. Technol., Dalian, China, 2019.

[18] H. Wang and Y. Song, “Secure cloud-based EHR system using attribute-based cryptosystem and blockchain,” J. Med. Syst., vol. 42, no. 8, p. 152, 2018.

[19] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, “Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain,” IEEE Access, vol. 8, pp. 192177–192191, 2020, doi: 10.1109/ACCESS.2020.3032680.

[20] S. Ghaffaripour and A. Miri, “Application of blockchain to patient-centric access control in medical data management systems,” in Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON), Oct. 2019, pp. 190–196, doi: 10.1109/IEMCON.2019.8936186.

[21] L. Zhang, L. Ling: Practical Medical Files Sharing Scheme

[22] R. Guo, H. Shi, D. Zheng, C. Jing, C. Zhuang, and Z. Wang, “Flexible and efficient blockchain-based ABE scheme with multi-authority for medical on demand in telmedicine system,” IEEE Access, vol. 7, pp. 88012–88025, 2019, doi: 10.1109/ACCESS.2019.2925625.

[23] H. Jin, C. Xu, Y. Luo, P. Li, Y. Cao, and J. Mathew, “Toward secure, privacy-preserving, and interoperable medical data sharing via blockchain,” in Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS), Dec. 2019, pp. 852–861, doi: 10.1109/ICPADS47876.2019.00126.

[24] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[25] G. R. Blakley, “Safeguarding cryptographic keys,” in Proc. Int. Workshop Manage. Requirements Knowl. (MARK), Jun. 1979, pp. 313–318.

[26] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT), 2005, pp. 457–473.

[27] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in Proc. Ann. Int. Conf. Theory Appl. Cryptograph. Techn., 2011, pp. 568–588.

[28] M. M. Madine, A. A. Battah, I. Yaqoob, K. Salah, R. Jayaraman, Y. Al-Hammadi, S. Pescis, and S. Ellahham, “Blockchain for giving patients control over their medical records,” IEEE Access, vol. 8, pp. 193102–193115, 2020, doi: 10.1109/ACCESS.2020.3032553.

[29] W. Ren, H. Fu, and H. Wang, “Multi-signature schemes based on bilinear pairing,” Comput. Eng. Des., vol. 28, no. 19, pp. 4637–4638, Oct. 2007.

[30] V. Buterin. (2014). Next-Generation Smart Contract and Decentralized Application Platform. [Online]. Available: https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf

JIYU TAO received the B.S. degree from the School of Information Science and Technology, Fudan University, Shanghai, China, in 2018, where he is currently pursuing the M.S. degree. His research interests include cryptography and blockchain technology.

LI LING received the M.S. degree from the School of Computer Science, Fudan University, in 1992. He is currently an Associate Professor with the School of Information Science and Technology, Fudan University. His research interests include information security, cryptography, and blockchain technology.

***