# A Privacy-preserving Medical Data Sharing Scheme Based on Blockchain

Guangquan Xu, *Member, IEEE*, Chen Qi, Wenyu Dong, Lixiao Gong, Shaoying Liu, *Fellow, IEEE*, Si Chen, Jian Liu, Xi Zheng, *Member, IEEE*

***Abstract*— With the increasing penetration of the Internet of things (IoT) into people's lives, the limitations of traditional medical systems are emerging. First, the typical way of handling sensitive information can easily lead to privacy disclosure. Second, the medical system is relatively isolated. It is difficult for one medical system to share data with another, and the scope of users' activities is limited within the system boundary. To solve these two problems, we propose a new privacy-preserving medical data-sharing scheme by introducing the authorization mechanism and attribute-based encryption (ABE) based on blockchain, which breaks system boundaries and realizes data sharing among several medical institutions. ABE is used to realize scalable access control. In addition, doctors can share their knowledge to diagnose users by introducing many-to-many matching, which means that patients' health data can be represented by multiple keywords and doctors' expertise can be represented by multiple interests. We provide the correctness and security analysis of our scheme and implement a prototype tool on Ethereum. The experimental results show that our scheme solves the contradiction between the privacy preservation of medical data and the necessity of data sharing.**

***Index Terms*— Attribute-based encryption, blockchain, privacy preserving, intelligent medical system.**

## I. Introduction

Advances in mobile communication and the internet have promoted the popularization of the Internet of Things (IoT). IoT connects many hardware devices and realizes data sharing on these devices to help improve monitoring and management. IoT has been widely used in smart transportation, smart retail, smart agriculture and other fields. For example, the agriculture-related data like soil properties and water level can be monitored in real time through IoT devices. Through a privacy-preserving data aggregation scheme based on ElGamal Cryptosystem in [51], we can make a balance between data sharing and privacy preservation. IoT has also made great achievements in the intelligent medical scenario. In an intelligent medical system, patients' physiological data are captured by sensors, processed centrally in the local gateway, and then sent to the medical service provider. Doctors can obtain patients' health data anywhere through the intelligent medical technology to facilitate remote medical treatment. Such an intelligent system realizes the interconnection between patients and doctors.

Although the intelligent medical system has changed the traditional medical treatment process and dissolved geographical restrictions associated with the traditional process, the existing intelligent medical system can only work independently and lacks cooperation. Patients can only initiate consultation within the system boundary. For example, patients who are in some specialized hospitals cannot be treated for diseases requiring other specialties, and some hospitals are not qualified enough to diagnose patients with more serious diseases. Township and community hospitals cannot achieve seamless information connection with large hospitals. This inability greatly limits the quality of medical treatment. To resolve it, we developed a new privacy-preserving medical data sharing scheme that can interconnect decentralized medical service providers to form a joint platform on the premise of mutual authorization between hospitals. It is easy to obtain expert opinions in real time and achieve data sharing when transferring from one hospital to another.

Blockchain can be regarded as a distributed recording ledger with the characteristics of anonymity, tamperability, auditability and autonomy. The smart contract (SC) running on it can avoid the interference of malicious users in the normal operation process, which is an effective solution for privacy-preserving medical data sharing. There are still many practical challenges to be solved when applying the blockchain to medical data sharing. The following are the major issues that interest us. 1) Patient medical data are highly sensitive and should be protected when uploading to the blockchain. Doctors and hospitals may compromise the patient's privacy without their permission for commercial interests. 2) Information sharing between medical institutions contradicts the high sensitivity of patient medical data.

In this paper, we construct a new privacy-preserving data-sharing scheme based on the blockchain for medical scenarios, which breaks system boundaries and realizes data sharing among several medical institutions. Patients can upload their own electronic medical records in privacy, the authorized relevant hospitals can conduct private keyword matching, the matched doctors can diagnose patients based on the provided electronic medical records, and the diagnosis results can be safely transmitted to the corresponding patients. Because the professional field of doctors has personalized attributes, this paper adopts attribute-based encryption to grant the data owner (i.e., the patient) the right to control the access of data. The matched doctors must decrypt the health data according to their own attributes before diagnosis. In addition, we introduce an authorized and revocable mechanism to ensure that doctors in authorized hospitals can obtain access to the data while the revoked doctor cannot obtain. Moreover, we use a zero-knowledge proof protocol to ensure the credibility of the hospital matching algorithm under privacy conditions.

Specifically, our key contributions in this scheme can be summa-

Manuscript submitted March 11, 2022. This work is partially sponsored by National Key RD Program of China (No. 2019YFB2101700), National Science Foundation of China (No. 62172297, No. 61902276), the Key Research and Development Project of Sichuan Province (No.2021YFSY0012), Tianjin Intelligent Manufacturing Special Fund Project (No. 20211097, No. 20201159), and Statistical Research Project of Shandong Province in 2022 ”The logic and path of artificial intelligence and Statistics Fusion” (KT060). (Corresponding author: Wenyu Dong; Lixiao Gong)

Guangquan Xu is with the School of Big Data, Qingdao Huanghai University, Qingdao 266000, China, and also with Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: losin@tju.edu.cn).

Chen Qi, Wenyu Dong, Lixiao Gong, and Jian Liu are with Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: luckychen468@163.com; dongwenyu@tju.edu.cn; glx_0826@tju.edu.cn; jianliu@tju.edu.cn).

Shaoying Liu is with the Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi Hiroshima 739-8511, Japan (e-mail:sliu@hiroshima-u.ac.jp).

Si Chen is with the Computer Science Department, West Chester University of Pennsylvania, West Chester, PA 19382 USA.

Xi Zheng is with the School of Computing, Macquarie University, Sydney, NSW 2109, Australia (e-mail:james.zheng@mq.edu.au).

rized as follows:

1) We construct a new data sharing scheme based on the blockchain for medical scenarios, which breaks system boundaries and realizes cross-hospital diagnosis.

2) We design an authorized and revocable mechanism to authorize or revoke the access to the data of doctors in hospitals to ensure flexible data access control.

3) Through searchable attribute-based encryption, doctors in authorized hospitals are allowed to generate search trapdoors according to their own interests and expertise, send query requests to the hospital, and realize many-to-many matching between the multiple keywords of the patient's health data and multiple interests of the doctor's expertise.

4) The zero-knowledge proof protocol ensures the credibility of hospitals' matching algorithm and patients' access to medical reports under privacy conditions.

The remainder of this paper is organized as follows. The related work is described in Section II. The preparatory work is included in Section III. Section IV introduces the system model, system procedure, and threat model. We review the details of the system implementation in Section V. We provide the correctness and security analysis of our scheme in Section VI. After the performance analysis given in Section VII, we draw conclusions in Section VIII.

## II. Related Work

Many fields are involved in this paper, such as blockchain technology, attribute-based encryption and intelligent medical systems. In this section, three main related topics are reviewed: 1) the application of blockchain in network scenes; 2) intelligent medical systems; and 3) attribute-based encryption.

### A. Application of the blockchain in a network scenario

As a database, the blockchain has been widely used in the field of decentralized networks [45], [56]. Blockchain technology has many characteristics [32]. The blockchain with decentralization and anonymity has become the core technology behind Bitcoin. In addition to its application in the financial field, the blockchain has also been applied to other fields, including smart transportation [29], [58], smart grid [31], and data auditing [25]– [27].

In the field of smart grid, malicious users may infer a user's private information from electricity consumption data. For this, Guan *et al.* introduced a data aggregation scheme based on blockchain to preserve users' privacy [31]. Users' identities are hidden in pseudonyms. A user can be associated with multiple pseudonyms to submit electricity consumption data. And in intelligent transportation system, Ning *et al.* constructed a crowdsensing framework based on the decentralized blockchain to realize key management in a distributed way [29], which makes a trade-off between minimizing the latency and maximizing the safety of the blockchain. Many scholars are committed to using blockchain to improve the security of Unmanned Aerial Vehicles (UAV). Ch *et al.* [55] presented a Blockchain Technology (BCT) based solution to better manage sensitive data and prevent data from being attacked.

However, the application of these schemes often requires specific scenarios, so it is inappropriate to apply them directly to intelligent medical scenarios. Both the patient's health data and the health report generated by the doctor should be protected. Therefore, for medical scenarios, we propose a scheme based on the blockchain that can realize data sharing on the premise of protecting user privacy.

TABLE I

Comparisons of known research work on the intelligent health system

| | [10] | [11] | [12] | [13] | [14] | [48] | Our work |
|---|---|---|---|---|---|---|---|
| $Cryptographic$ | ✓ | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| $Noncryptographic$ | - | ✓ | - | - | - | - | - |
| $Blockchain-based$ | - | - | ✓ | ✓ | ✓ | - | ✓ |
| $Cloud-based$ | ✓ | ✓ | - | - | - | ✓ | - |
| $Searchable$ | - | - | - | ✓ | - | ✓ | ✓ |
| $Joint$ | - | ✓ | - | - | - | - | ✓ |

### B. Privacy preservation in intelligent health systems

In this era of information explosion, data means resources, but the more such resources, the greater the possibility of privacy leakage. People try to illegally occupy and exploit resources, and the privacy security has not been ignored. Many scholars are committed to enhancing the privacy protection and reducing illegal appropriation of resources [1], [16]– [18], [57]. The intelligent medical system is no exception. The transformation from the traditional health care system to the electronic health care system has made clinical data easier and faster to access. However, inevitably, patients' privacy concerns cannot be ignored. The patient's body data are personal and sensitive. Direct exposure to the shared cloud environment will eliminate data privacy [3], which will not only affect relevant laws and regulations but also have a serious economic and social impact on the patient. Therefore, it is an urgent problem to introduce a strong privacy protection mechanism into the whole intelligent medical system. A mechanism to reduce the linkability between patients and medical records is proposed by Li *et al.* [7]. Hupperich *et al.* [10] mentioned out that the existing privacy protection is either too strict and requires patients to be available to authorize access to medical records or is insufficient and does not truly realize privacy protection. Therefore, it is necessary to provide more flexibility for whole system to ensure that doctors can access medical records without the presence of patients. Abbas *et al.* [4] mentioned out that the privacy preserving methods commonly used in intelligent medicine are divided into two categories: 1) cryptographic approaches, which use specific encryption primitives to reduce privacy risks, and 2) noncryptographic approaches, which mainly adopt a policy-based infrastructure, to standardize the access control of data. A new access control mechanism is provided [11], which is a noncryptographic approach to support the fine-grained sharing of electronic health records from different medical service providers in the cloud by implementing the access control policies specified by patients. Cui *et al.* [4] combined attribute-based encryption with keyword search in a cloud storage system that keeps personal health records. However, it requires the cloud server to be honest, and this scheme only consider the scene where patients are treated for diseases in a single medical institution rather than multiple institutions. This limitation is not compatible with the requirement to break the geographical restrictions of medical institutions.

With scalability, flexibility, and some economic reasons, the health care data sharing method based cloud is very popular, but when patients upload data to the cloud server, the potential risk of the cloud worries data owners. With the popularity of Bitcoin, the blockchain has gradually become a better solution to security and privacy problems by virtue of its data integrity and distributed storage characteristics. Esposito *et al.* [5] mentioned out that using the blockchain to provide secure health care data management can ensure that 1) A credible authority is not necessary to reach an agreement and there is no need to worry about a single point of failure; 2) data owners can control their own data; and 3) electronic medical records are stored on the blockchain in a distributed manner. Aslam *et al.* [53] proposed a framework to trace the contact of the general public, which keeps people safe from infected in a distributed way. This scheme keeps the anonymity of data but the data owner cannot determine the access permission of his or her data. Aiming at the problem that patients cannot control their own data and adhering to the principle that medical data should not be controlled by untrusted third parties, a health care data gateway (HDG) architecture is designed [12], which combines traditional databases and gateways to manage personal medical data stored on the blockchain. The introduction of secure multiparty computing enables third parties to process data without compromising patients' privacy. Wang *et al.* [52] proposed a lightweight and reliable authentication protocol to achieve data sharing between sensor nodes and medical professional. Gateway nodes are not centralized and credible. They form a blockchain network and over-centralized server problem is solved. Trusted third parties are not necessary for data owners to know who is accessing their data and how the data will be used. Privacy protection focuses on what data objects are used for what purposes.

However, with the significant increasing of the number of users, the storage pressure on the blockchain is gradually increasing. An efficient medical data sharing scheme-based session is proposed, and a chained digest creation method is designed to realize efficient data sharing by combining blockchain, digest chain and structured peer-to-peer network technology so that patients can better control their medical data and reduce information fragments [13]. However, there are limitations to this kind of data sharing. One is that data mobility is not considered. If patients migrate from one city to a new city or from one hospital to another, accessing the data of another hospital involves cross-organizational data sharing, which will cause many additional expenses. Another limitation is that data access needs to be manually specified by the patient, which will increase the operation overhead and the access delay of the requester. This delay should be avoided as much as possible in the scenario of medical diagnosis. A better solution is to realize automatic data sharing through authorization and other mechanisms. In 2021, Lee *et al.* [13] proposed a blockchain-based medical data preservation scheme. In the proposed scheme, only the authorized parties can access the data. However, this work requires that the key should be issued once for each authorization and the efficiency needs to be improved. Hoai *et al.* [14] started from data writing to reduce the storage pressure on the blockchain. Unlike the traditional scheme, they considered how to reduce the storage pressure after writing the original data into the blockchain. Initially, they considered whether it was necessary to write all the data into the blockchain, filter the data from the sensor and then write them into the blockchain. This approach can greatly reduce the storage pressure on the blockchain. An electronic health system based on the blockchain is proposed by Zou *et al.* for medical data sharing and privacy protection [15]. Patient electronic medical records are stored on special key blocks and microblocks to achieve rapid retrieval. However, the communication overhead of this scheme is so large that the application scenario will be subject to many restrictions.

A comparison of the main studies is shown in Table I. Searchable means that queries over ciphertext are supported. From Table I we can see that almost all works use cryptographic primitives to preserve the privacy. Researchers are increasingly turning to blockchain as a trusted alternative to the cloud. Joint indicates whether it supports data sharing across multiple healthcare institutions. Most works consider privacy protection in intelligent medical systems, but few works consider how to realize data sharing among several medical institutions. In the social health care network, patients' data often exist at a node. From a clinical point of view, data sharing and integration between different medical service providers are necessary for medical diagnosis. Therefore, we urgently need to realize an effective and safe data sharing mechanism among multiple medical institutions [4].

### C. Attribute-based encryption

In this section, we introduce the basic mechanism of attribute based encryption (ABE) in detail. ABE is a relatively new encryption mechanism which does not require information interchange between data owners and users and is suitable for the one-to-many distribution. The user's private key and ciphertext are constructed based on the attribute set and access policy. As long as the attribute set matches the policy, plaintext can be obtained.

Many studies have sought to optimize the algorithm and improve the efficiency. Jin *et al.* [19] proposed a secure and lightweight data access control scheme. Most of the computation operations are performed by the cloud and the computing overhead of users is greatly reduced. Lewko *et al.* [20] designed two functional encryption schemes including an attribute-based encryption and a predicate encryption for inner product predicates, which are fully secure systems. Xu *et al.* [2] proposed a privacy-enhanced access control mechanism. Different schemes are implemented according to whether the attribute belongs to the sensitive attributes, and finally the comprehensive decision is made. Sensitive data should not be uploaded directly before outsourcing but should be encrypted; otherwise, user privacy will be compromised. However, the encrypted data affect the efficiency of data querying [21]. Searchable encryption allows users to search directly on ciphertext using keywords without decrypting data [22], which solves the contradiction between data confidentiality and searchability. In searchable encryption, users are often allowed to generate a trapdoor according to their own interests and then match the trapdoor with keywords. Attribute-based searchable encryption successfully combines the advantages of the two. Due to the high computational cost of anonymity, Han *et al.* introduced a weak anonymity feature to hide users' identity. Based on this weak anonymity, they proposed a general transformation from attribute-based encryption to attribute-based encryption with keyword search and constructed a specific attribute private key-policy ABE scheme, which allows multiple users to flexibly search remote encrypted data [23]. However, most existing attribute-based searchable encryption approaches support only one-to-one or one-to-many retrieval, and few approaches support many-to-many retrieval. In our design, we realize many-to-many retrieval based on attribute-based encryption.

## III. Preliminaries

In this part, we explain the basic concepts and relevant knowledge that will be used to discuss our proposed system in the next section. We first give some introduction on bilinear pairings, access structure, and linear secret sharing schemes. All of this will be used to construct the algorithm. In our scheme, the attributes will play the role of parties defined in the access structure. In addition, we introduce the zero-knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) to ensure the credibility of our scheme.

### A. Bilinear Pairings

Bilinear mapping can be described by quintuples $(p, G_1, G_2, G_T, e)$. Let $G_1$, $G_2$, and $G_T$ be cyclic groups of order $p$, and let $p$ be the prime number. If mapping $e : G_1 \times G_2 \to G_T$ meets the following properties:

*1) Bilinear:* For any $g_1 \in G_1$, $g_2 \in G_2$, and $a, b \in Z_p$, there is $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;

*2) Non degenerative:* For any $g_1 \in G_1/\{1\}$, there is always $g_2 \in G_2$, so that $e(g_1, g_2) \neq 1$. Here, 1 represents the unit element of the $G_T$ group;

*3) Computability:* There is an effective polynomial time algorithm to calculate the value of $e(g_1, g_2)$ for any $g_1 \in G_1$ and $g_2 \in G_2$,

We call mapping $e$ a bilinear mapping from $G_1 \times G_2$ to $G_T$.

### B. Access Structure and Linear Secret Sharing Schemes

*Definition 1(Access Structure):* Let $P = \{P_1, ..., P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1,...,P_n\}}$ is monotone $\forall B, C$: $B \in A$ and $B \subseteq C \Rightarrow C \in A$. An access structure is a monotone collection $A$ of nonempty subsets of $\{P_1, ..., P_n\}$, i.e., $A \subseteq 2^{\{P_1,...,P_n\}} \backslash \emptyset$. The sets in $A$ are called authorized sets, while the sets not in $A$ are called unauthorized sets.

*Definition 2(Linear Secret Sharing Schemes(LSSS)):* Let $P$ be a set of parties. Let $M$ be a matrix of size $l \times n$. Let $\rho : 1, ..., l \rightarrow P$ be a function that maps a row to a party for labeling. A secret sharing scheme $\Pi$ over a set of parties $P$ is a linear secret-sharing scheme over $Z_p$ if

1) the shares of each party from a vector over $Z_p$;

2) there exists a matrix $M$ with $l$ rows and $n$ columns, called the share-generating matrix, for $\Pi$. For $x = 1, ..., l$, the $x$-th row of matrix $M$ is labeled by a party $\rho(i)$, where $\rho : 1, ..., l \rightarrow P$ is a function that maps a row to a party for labeling. Considering that the column vector $\vec{v} = (\mu, r_2, ..., r_n)^{\mathrm{T}}$, where $\mu \in Z_p$ is the secret to be shared and $r_2, ..., r_n \in Z_p$ are randomly chosen, then $M\vec{v}$ is the vector of $l$ shares of the secret $\mu$ according to $\Pi$. The share $(M\vec{v})_i$ belongs to party $\rho(i)$.

### C. zk-SNARK

Zero-knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) is a novel form of zero knowledge cryptography [47]. There are two sides: the prover and the verifier. Without disclosing the specific data, the prover can make the verifier believe the authenticity of the data. The following steps:

*1) Setup:* Let $\vec{x}$ represent the common knowledge, and $\vec{\omega}$ represent the private witness. This step outputs the proving key and verification key, which are used for proof generation and proof verification, respectively, for the language $L_T = \{\vec{x} \mid \exists\vec{\omega}, s.t., C(\vec{x}, \vec{\omega}) = 1\}$.

*2) Prove:* The prover takes the proving key, the common knowledge $\vec{x} \in L_T$, and the private value $\vec{\omega}$ as inputs and obtains a proof (a.k.a. proof-of-knowledge) $\pi$ by running the $Prover$ algorithm.

*3) Verify:* The verifier takes the verification key, $\vec{x}$, and $\pi$ as inputs to verify the proof by running the $Verifier$ algorithm. Return 1 if the verification succeeds and 0 otherwise.

## IV. System Design

In this part, the system model, system procedure, and threat model are described.

### A. System Model

As shown in Fig. 1, our model involves five entities: patients, doctors, hospitals, the key generation center (KGC), and blockchain platforms. Symbols used in this scheme are shown in Table II.

*1) Patients:* As the requesters of medical consultation system services, patients can upload their own physiological data or medical records to the hospital. Patients can obtain the information from the hospital to know their physical status.

TABLE II
GLOSSARY

| Symbol | Definition |
|---|---|
| $H$ | set of hospitals |
| $sk_{id}, pk_{id}$ | identity key pair of user $id$ |
| $rl$ | revocation list |
| $tm$ | time mark |
| $st$ | system state mark |
| $A_{id}$ | attributes of use $id$ |
| $(M, \rho)$ | LSSS access matrix and its mapping function |
| $CT_{HD}$ | ciphertext of the health data owned by patient |
| $SymKey/CT_{Sym}$ | symmetric key for data/ciphertext of $SymKey$ |
| $kw_i$ | $i$-th keyword |
| $inst_i$ | $i$-th interest |
| $attk_{id}$ | attribute key of user $id$ |
| $updk_{tm}$ | updating key when the time mark is $tm$ |
| $prek_{id,\ tm}$ | predecryption key of user $id$ when the time mark is $tm$ |
| $SemiCT$ | ciphertext after predecryption |
| $CT_{HR}$ | ciphertext of health report |

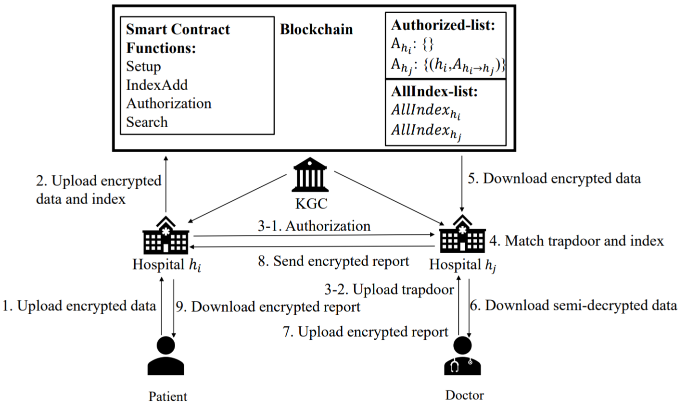


Fig. 1. The data flow of the system.

*2) Doctors:* Doctors can match the data uploaded by patients according to their own interests, complete the diagnosis after successful matching, and transmit the diagnosis results to the corresponding patients.

*3) Hospitals:* All patients and doctors belong to a hospital. Each hospital has corresponding patients and doctors. Doctors provide services to patients through hospitals. As service providers, hospitals are responsible for matching patients and doctors.

*4) KGC:* The KGC is honest and generate the predecryption key according to the public key and the attributes of the doctor. The leaving doctor will be added to the revocation list by the KGC.

*5) Blockchain platform:* Blockchain and smart contracts connect all hospitals to form a joint medical consultation system. All hospitals share data through the blockchain.

### B. System Procedure

The data flow in this scheme is shown in Fig. 1. The procedure of the whole scheme is as follows.

*1) System Initialization:* The whole system is initialized. $Setup(1^{\lambda}) \rightarrow (pp, msk, rl, st)$ is run by the KGC. Given a security parameter $\lambda$, the public parameter $pp$, the master private key $msk$, a revocation list $rl$, and a system state $st$ are output by the algorithm.

*2) User Registration and Revocation:* Anyone who wants to join the system needs to run $UserKG(pp, id) \rightarrow (sk_{id}, pk_{id})$. The algorithm generates its identity key pair according to the id of the user. $Revoke(id, tm, rl, st) \rightarrow rl$ will be run to revoke the user by updating the revocation list. The algorithm takes the id of the revoked user, time mark $tm$, and the system state $st$ as inputs and a new revocation list is output.

*3) Encryption:* If a user wants to upload his or her health data, the data must be encrypted by running a symmetric algorithm using SymKey and the data owner can obtain $CT_{HD}$. Then, $Encrypt(pp, (M, \rho), tm, SymKey) \rightarrow CT_{Sym}$ is run by the data owner to hide the Symkey. Then, the $CT_{HD}$ and $CT_{Sym}$ will be sent to hospital $h_i$, where the patient visits.

*4) Index Generation:* $IndexBuild(pp, \{H(kw)\}) \rightarrow Index$ is run by the hospital where the patient wants to be treated. $kw$ is a keywords set selected by the patient from the data, and $H$ is a hash function. Then, the index is created and will be uploaded to the blockchain.

*5) Authorization:* If a hospital does not have enough doctors or the patients' disease exceeds the hospital's diagnostic ability, the hospital will authorize access to the patient's data to other hospitals that have the ability to diagnose the patient. The hospital will generate $A_{h_i \rightarrow h_j}$, which means $h_i$ authorizes to $h_j$ and publishes it to the blockchain.

*6) Search:* The authorized hospital can obtain the correct index using the provided $A_{h_i \rightarrow h_j}$.

*7) Key Generation:* There are three steps to complete. First, $AttKeyGen(pp, msk, id, pk_{id}, A_{id}, st) \rightarrow (attk_{id}, st)$ is run by KGC according to the patient's attributes $A_{id}$. Then, $UpdKeyGen(pp, msk, tm, rl, st) \rightarrow (updk_{tm}, st)$ is run by KGC according to the revocation list to ensure that the revoked doctor cannot obtain the key $updk_{tm}$. $PreKeyGen(pp, id, attk_{id}, updk_{tm}) \rightarrow prek_{id,tm}$ is finally executed taking the $attk_{id}$ and $updk_{tm}$ generated in the first two steps as input. The $prek_{id,tm}$ will be used to predecrypt.

*8) Trapdoor Generation:* If a doctor wants to obtain the patients' data, he or she needs to run $TrdGen(pp, \{H(inst)\}, m_1) \rightarrow Trapdoor$. This step takes the hash of the doctor's interest $inst$ and the number of keywords $m_1$ as inputs. Note that a doctor can correspond to multiple interests. Then $Trapdoor$ will be sent to the hospital where the doctor works for matching.

*9) Match:* $Match(Trapdoor, Index) \rightarrow Addr/\perp$ is run by the hospital to help doctors find the index corresponding to their interest. If the condition that the doctors' interest set is a subset of the keyword set is satisfied, the match is successful. If the match succeeds, it returns the address of the data. Otherwise, it returns $\perp$. If multiple doctors matched the same set of keywords, the doctor with more interests is selected to complete the diagnosis.

*10) Decryption:* Two steps are required to obtain the plaintext of $SymKey$. First, $PreDec(pp, id, A_{id}, prek_{id,tm}, CT_{Sym}, tm) \rightarrow SemiCT/\perp$ is run by the hospital to transform $CT_{Sym}$ to $SemiCT$. Then the doctor runs $Dec(pp, sk_{id}, SemiCT) \rightarrow SymKey/\perp$. If the attributes of the doctor meet the requirement, the symmetric key can be decrypted. Using $SymKey$ and $CT_{HD}$, the doctor can obtain the plaintext of the health data.

*11) Results Return:* The doctor can then diagnose the patient. The health report is encrypted to $CT_{HR}$ using the RSA encryption algorithm [46]. Finally, $CT_{HR}$ is sent to the patient by hospitals. The patient can decrypt it using the private key to obtain the health report.

### C. Threat Model

In our scheme, we make the following assumptions and consider the following threats.

*1) KGC:* The KGC is entirely trustworthy to accomplish the key distribution and user revocation.

*2) Blockchain:* The whole blockchain mechanism is completely honest. All the consensus nodes work together to store data. The authorization information and index information are safely and reliably stored on the blockchain. All the consensus nodes complete authorization, index add, and search according to the process defined in the smart contract.

*3) Doctor:* In our scheme, we assume that the doctors strictly implement the industry norms, submit valid interests, complete the diagnosis for users honestly, and truthfully report the diagnosis results to the hospital.

*4) Patient:* We assume that the patients submit valid health data and encrypt the data honestly. However, the patient may be malicious in the sense that he or she modifies the diagnosis results to frame the doctor.

*5) Hospital:* The hospital may be untrustworthy, for instance, if the hospital matches unqualified doctors for potential benefits.

## V. Implementation Of Our Design

In this part, we introduce the SC and the construction of our system in detail.

### A. Smart Contract Functions

There are four functions in the SC as shown in Algorithm 1.

**Algorithm 1:** SC functions

```
Function Setup($H$):
    for each $h_i$ in $H$ do
        $A_{h_i} \leftarrow \emptyset$ ;
    $AllIndex \leftarrow \emptyset$ ;

Function IndexAdd($Index$):
    for each $< E_{t_1}, (I_1, I_2) >$ in $Index$ do
        $\hat{E_{t_1}} \leftarrow G_1(E_{t_1})$;
        $\hat{I_1} \leftarrow G_2(E_{t_1}) \oplus I_1$;
        $\hat{I_2} \leftarrow G_2(E_{t_1}) \oplus I_2$;
        $AllIndex[\hat{E_{t_1}}] \leftarrow (\hat{I_1}, \hat{I_2})$;

Function Authorization($h_i, h_j, A_{h_i \rightarrow h_j}$):
    $A_{h_j}[h_i]$.add($A_{h_i \rightarrow h_j}$);

Function Search($h_i, h_j, \omega$):
    $Res \leftarrow \emptyset$;
    for each $A$ in $A_{h_i}[h_j]$ do
        $E_{t1} \leftarrow A^{\omega}$;
        if $G_1(E_{t_1}) \in AllIndex$ then
            $\hat{I_1}, \hat{I_2} \leftarrow AllIndex[\hat{E_{t1}}]$;
            $Res \leftarrow Res + \{\hat{E_{t_1}}, \hat{I_1}, \hat{I_2}\}$
```

*1) Setup:* This function is executed in the phase of system initialization. For each medical institution in the system, set its authorization list to empty. The index set in the whole system is initialized to an empty set.

*2) IndexAdd:* This function is executed in the phase of index generation. For each index generated by hospital $h_i$, SC will add it to the blockchain. To prevent other hospitals from seeing the index information, the index cannot be uploaded directly. The hospital uploads $G_1(E_{t_1})$ instead of $E_{t_1}$ where $G_1$ is a pseudo-random function. Let $\hat{I_1} = G_2(E_{t_1}) \oplus I_1$ and $\hat{I_2} = G_2(E_{t_1}) \oplus I_2$ where $G_2$

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3203577

is a pseudorandom function. Then, $\hat{I}_1$ and $\hat{I}_2$ are uploaded instead of $I_1$ and $I_2$.

*3) Authorization:* All indexes authorized by $h_i$ to $h_j$ are added to the collection $A_{h_i}[h_j]$.

*4) Search:* If a hospital $h_j$ wants to obtain access to the data, it traverses the authorized list $A_{h_i}[h_j]$ to calculate the correct $E_{t_1}$. The complete index can be obtained by accessing the index array $AllIndex$.

### B. Construction of Our Design

In this part, we introduce the implementation details of the scheme.

*1) System Initialization:* In the initialization phase, the KGC generates the master secret key and corresponding parameters. Then, the task index and the authorization list between different hospitals are initialized by calling the SC function SC. Specifically, the KGC randomly chooses a group $G_1$ of prime order $p$, and $g \in G_1$ is the generator. We define a bilinear map $e : G_1 \times G_1 \to G_T$. It randomly chooses $u$, $h$, $u_0$, $h_0$, $w$, $v$, $f \in G_1$, $\alpha, \theta \in Z_p$. Revoked users are stored in the empty list $rl$ which is initialized in this step. A binary tree $BT$ that has at least N leaf nodes responding to N users is constructed [24]. The root of the tree is denoted by $root$. Each node is represented by a unique code. The $st$ is the state of the binary tree. Each node $x$ corresponds to a $g_x \in Z_p$. To map an element $y$ in $Z_p$ to an element in $G_1$, two functions are defined: $F_1(y) = u^y h$ and $F_2(y) = u_0^y h_0$. The public parameter and master secret key are $pp = (g, f, w, v, u, h, u_0, h_0, e(g, g)^{\alpha}, g^{\alpha}, g^{\theta})$ and $MSK = (\alpha, \theta)$.

*2) User Registration and Revocation:* $UserKG$ is run to obtain an identity key pair $(sk_{id}, pk_{id}) = (\beta_{id}, g^{\beta_{id}})$ where $\beta \in Z_p$ is a random number. In addition, each newly registered user is bound to an unused leaf node $\theta$ in binary tree $BT$. If a user needs to be revoked, all the nodes associated with the user will be added to the revocation list in the form of $(x, tm)$, where $x$ is the node.

*3) Encryption:* The health data are encrypted to $CT_{HD}$ using a symmetric algorithm by the patient. Then, $Encrypt$ is run. Let $A$ be the attribute set. There are $k$ attributes in total. Let $(M, \rho)$ be an access structure where $M$ be a $r \times c$ matrix and $\rho$ be a mapping that maps the $i$-th row of the matrix $M$ to an attribute $A_{\rho(i)}$ in $A$. It defines a vector $\vec{v} = (s, r_2, r_3, ..., r_c)^{\mathrm{T}} \in Z_p^c$, where $s \in Z_p$ is a secret to be shared and $r_2, ..., r_c \in Z_p$ are randomly chosen. Then, $v_j = (M\vec{v})_j$ for $j \in [1, r]$. It randomly chooses $\mu_1, \mu_2, ..., \mu_r \in Z_p$. The $SymKey$ is encrypted to $CT_{Sym} = ((M, \rho), tm, \{C_{0,j}, C_{1,j}, C_{2,j}\}_{j \in [1,r]}, C_3, C_4, C_5)$ where

$$\begin{aligned} C_{0,j} &= w^{v_j} v^{\mu_j}, \\ C_{1,j} &= F_1(A_{\rho(j)})^{-\mu_j}, \\ C_{2,j} &= g^{\mu_j}, \\ C_3 &= e(g, g)^{\alpha s} \cdot SymKey, \\ C_4 &= g^s, \\ C_5 &= F_2(tm)^s. \end{aligned} \tag{1}$$

*4) Index Generation:* $KW = \{kw_1, kw_2, ..., kw_{m_1}\}$ is a keyword set selected by the patient. An $m_1$ degree polynomial $P(x) = p_{m_1} x^{m_1} + p_{m_1 - 1} x^{m_1 - 1} + ... + p_0$ is constructed that meets the requirement that $H(kw_1)$, ..., $H(kw_{m_1})$ are the $m_1$ roots for $P(x) = 1$. In addition, a random number $b_1$ is chosen. Hospital $h_i$ builds $Index = (I_1, I_2, \{E_{t_1}\}_{t_1 \in [0, m_1]})$ based on keywords by running $IndexBuild$ where

$$\begin{aligned} I_1 &= g^{b_1}, \\ I_2 &= g^{\theta b_1}, \\ E_{t_1} &= e(g, f)^{p_{t_1}} e(g, g)^{p_{t_1} \alpha b_1}. \end{aligned} \tag{2}$$

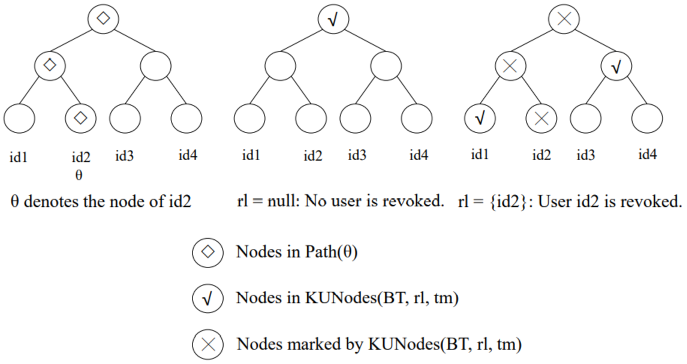

id1
id2
θ
id3
id4
id1
id2
id3
id4
id1
id2
id3
id4
θ denotes the node of id2
rl = null: No user is revoked.
rl = {id2}: User id2 is revoked.
◇ Nodes in Path(θ)
√ Nodes in KUNodes(BT, rl, tm)
× Nodes marked by KUNodes(BT, rl, tm)

Fig. 2. An example of a binary tree.

$m_1$ is the number of keywords. Hospital $h_i$ sends $CT_{Sym}$, $CT_{HD}$, and $Index$ to the blockchain. Then, the SC function SC. IndexAdd is run to add the index to the index set $AllIndex$.

*5) Authorization:* $h_j$ generates a parameter $\omega$ that will be sent to $h_i$ through a private channel if $h_i$ wants to authorize $h_j$ so that doctors in $h_j$ can also diagnose patients in $h_i$. Then, $A_{h_i \to h_j}$ will be generated and sent to the blockchain. Then, the SC function SC. Authorization is run. $A_{h_i \to h_j}$ are as follows:

$$A_{h_i \to h_j} = \{E_{t_1}^{1/\omega}\}_{t_1 \in [0, m_1]}. \tag{3}$$

*6) Search:* Only $h_j$ can obtain the correct $\{E_{t_1}\}_{t_1 \in [0, m_1]}$ by calculating the following:

$$\{E_{t_1}\}_{t_1 \in [0, m_1]} = A_{h_i \to h_j}^{\omega}. \tag{4}$$

The hospital $h_j$ can obtain the indexes authorized to it by running the SC. Search.

*7) Key Generation:* A predecryption key will be generated through the following three stages. Let $A_{id}$ be the attributes of user $id$. The length of $A_{id}$ is $k_1$, and $A_{id,i}$ is the $i$-th attribute. First, $AttKeyGen$ is run by the KGC. For user $id$, $\theta$ is the corresponding leaf node. Define $I = Path(\theta)$ as the path from node $\theta$ to the node $root$ in the binary tree. We provide a graphical description of $Path(\theta)$ in Fig. 2. Each node $x \in Path(\theta)$, randomly chooses $r_x, r_{x,1}, ..., r_{x,k_1} \in Z_p$. The attribute key $attk_{id} = \{x, \{P_{x,1,i}, P_{x,2,i}\}_{i \in [1, k_1]}, P_{x,3}, P_{x,4}\}_{x \in Path(\theta)}$ is output, where

$$\begin{aligned} P_{x,1,i} &= g^{r_{x,i}}, \\ P_{x,2,i} &= F_1(A_{id,i})^{r_{x,i}} \cdot v^{-r_x}, \\ P_{x,3} &= pk_{id}^{\alpha} / g_x \cdot w^{r_x}, \\ P_{x,4} &= g^{r_x}. \end{aligned} \tag{5}$$

Then, $UpdKeyGen$ is run according to the revocation list. All the leaf nodes associated with users in the revocation list and all ancestral nodes are noted as revoked. Then the unrevoked children nodes of revoked nodes are added to the set $J = KUNodes(BT, rl, tm)$. Nodes set with the minimum number to be updated is calculated by the algorithm $KUNodes(BT, rl, tm)$ so that revoked users cannot obtain the key. We provide a graphical description about $KUNodes(BT, rl, tm)$ in Fig. 2. Then, for each $x \in KUNodes(BT, rl, tm)$, $s_x \in Z_p$ is chosen randomly. The updating key $updk_{tm} = \{x, Q_{x,1}, Q_{x,2}\}_{x \in KUNodes(BT, rl, tm)}$ is output, where

$$\begin{aligned} Q_{x,1} &= g_x \cdot F_2(tm)^{s_x}, \\ Q_{x,2} &= g^{s_x}. \end{aligned} \tag{6}$$

The attribute key and the updating key are both used in $PreKeyGen$. If $I \cap J$ is null, it returns $\perp$. For all $x \in I \cap J$, it randomly

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3203577

chooses $r'_x$, $r'_{x,1}$,...,$r'_{x,k_1}$, $s'_x \in Z_p$. From the first two steps, we know $attk_{id} = \{x, \{P_{x,1,i}, P_{x,2,i}\}_{i\in[1,k_1]}, P_{x,3}, P_{x,4}\}_{x\in Path(\theta)}$ and $updk_{tm} = \{x, Q_{x,1}, Q_{x,2}\}_{x\in KUNodes(BT,rl,tm)}$. It outputs $prek_{id,tm} = \{x, \{tk_{1,i}, tk_{2,i}\}_{i\in[1,k_1]}, tk_3, tk_4, tk_5\}_{x\in I\cap J}$, where

$$\begin{aligned} tk_{1,i} &= P_{x,1,i}\cdot g^{r'_{x,i}} = g^{r_{x,i}+r'_{x,i}}, \\ tk_{2,i} &= P_{x,2,i}\cdot F_1(A_{id,i})^{r'_{x,i}}\cdot v^{-r'_x} \\ &= F_1(A_{id,i})^{r_{x,i}+r'_{x,i}}\cdot v^{-(r_{x,i}+r'_{x,i})}, \\ tk_3 &= P_{x_3}\cdot Q_{x,1}\cdot w^{r'_x}\cdot F_2(tm)^{s'_x} \\ &= pk_{id}^{\alpha}\cdot w^{(r_x+r'_x)}\cdot F_2(tm)^{s_x+s'_x}, \\ tk_4 &= P_{x,4}\cdot g^{r'_x} = g^{r_x+r'_x}, \\ tk_5 &= Q_{x,2}\cdot g^{s'_x} = g^{s_x+s'_x}. \end{aligned} \tag{7}$$

*8) Trapdoor Generation:* If a doctor wants to traverse the data uploaded by patients according to their own interests $INST = \{inst_1, inst_2, ..., inst_{m_2}\}$, a trapdoor must be generated. The doctor randomly chooses random numbers $\kappa$, $b_2$, and $r \in Z_p$. The doctor constructs the $Trapdoor = \{T_1, T_2, T_3, T_4, \{T_{5,t_1}\}_{t_1\in[0,m_1]}\}$ as follows:

$$\begin{aligned} T_1 &= g^{\frac{\alpha b_2}{\beta+r}}, \\ T_2 &= \beta + r, \\ T_3 &= \kappa_2 b_2 m_2^{-1}, \\ T_4 &= e(g,f)^{b_2}, \\ T_{5,t_1} &= \kappa_2^{-1}\sum_{t_2=1}^{m_2} H(inst_{t_2})^{t_1}. \end{aligned} \tag{8}$$

*9) Match:* The hospital $h_j$ runs $Match$ to help doctors find the data corresponding to their interests. First, check if the attributes of the doctor satisfy the access structure. If they don not meet, it returns $\perp$. Second, it will judge whether the following equation holds or not when $Index = (I_1, I_2, \{E_{t_1}\}_{t_1\in[0,m_1]})$ and $Trapdoor = \{T_1, T_2, T_3, T_4, \{T_{5,t_1}\}_{t_1\in[0,m_1]}\}$. If it holds, the address $Addr$ will be sent to the hospital.

$$T_4 e(T_1, I_1^{T_2}) = \prod_{t_1=0}^{m_1} E_{t_1}^{T_3 T_{5,t_1}}. \tag{9}$$

In the process of matching, the hospital needs to provide a zero-knowledge proof to ensure that the match is indeed completed with the authorized index that meets the doctor's interest to prevent the hospital from cheating. Let $Index = (\{E_{t_1}\}_{t_1\in[0,m_1]}, I_1, I_2)$ represents the private witness, $Trapdoor$ and $A_{h_i\to h_j}$ are all common knowledge for the following language $L_T$ = $\{Trapdoor$, $A_{h_i\to h_j} \mid \exists\ Index = (\{E_{t_1}\}_{t_1\in[0,m_1]}, (I_1, I_2)),\ s.t.,$ $Match(Trapdoor, Index) = 1\ \wedge A_{h_i\to h_j} = \{E_{t_1}^{1/\omega}\}_{t_1\in[0,m_1]}$ $\}$. The result proves the algorithm yields a proof $\eta_{Match}$ for the statement $A_{h_i\to h_j} \in L_T$ also for the proof-of-knowledge of $Index$. The patient can run the verifying algorithm of zk-SNARK $Verifier$ on $\eta_{Match}$, and outputs the bit $d \in \{0,1\}$.

*10) Decryption:* Hospital $h_j$ runs $PreDec$ to reduce the computational pressure on doctors. Define an empty set $R$ and $\epsilon = (1,0,...,0)^c$. For each $j \in [1,r]$, if $A_{\rho(j)} \in A_{id}$, $j$ will be added to $R$. If $A_{id}$ meets the access structure, which means that attribtues in $R$ can recover the secret $s$, there will be a set of constants $\{\omega_j \in Z_p\}_{j\in R}$ satisfying $\sum_{j\in R}\omega_j M_j = \epsilon$. It outputs $SemiCT = (C'_0, C_3)$ where

$$C'_0 = \frac{\prod_{j\in R}(e(C_{0,j}, tk_4)e(C_{1,j}, tk_{1,j})e(C_{2,j}, tk_{2,j}))^{\omega_j} e(C_5, tk_5)}{e(C_4, tk_3)}. \tag{10}$$

Then the hospital sends the $SemiCT$ to the doctor, and the doctor obtains the $SymKey$ by calculating the equation $SymKey = (C'_0)^{1/\beta_{id}}\cdot C_3$.

*11) Results Return:* If a doctor completes the diagnosis, he/she will encrypt the health report using the RSA encryption algorithm. Then $CT_{report}$ will be sent to hospital $h_j$. In addition, $h_j$ will send it to $h_i$ by a private channel. Finally, the patient can obtain the report.

In the process of transmitting the results, the patient needs to provide a zero-knowledge proof to ensure that the plaintext is obtained by using the correct private key to decrypt. Let the private key $sk$ as the private witness. And the patient generates a zero-knowledge proof $\eta_{HR}$. The public key $pk$, the ciphertext $CT_{HR}$, and the plaintext $HR$ are input, and the zk-SNARK proving algorithm is run to compute the proof for the language $L_T$ = $\{CT_{HR}, HR,$ $pk \mid \exists sk,\ s.t.,\ pair(sk, pk) = 1 \wedge Decrypt(sk, CT_{HR}) = HR\}$ where $Decrypt$ is the decryption algorithm in the RSA encryption algorithm. The doctor can run the verifying algorithm of zk-SNARK $Verifier$ on $\eta_{HR}$ and outputs the bit $d \in \{0,1\}$.

## VI. Correctness and Security Analysis

### A. Correctness Analysis

We always assume that the result of decryption of ciphertext in our scheme is equal to plaintext and that the $Match$ algorithm can realize the match between doctors' multiple interests and multiple keywords. We provide the correctness analysis of our scheme in the following theorem.

*Theorem 6.1:* The authorized doctor can decrypt correctly and obtain the correct health data.

*Proof:* It is clear that $\sum_{j\in R}\omega_j v_j = s$ under the condition of $\sum_{j\in R}\omega_j M_j = \epsilon$. Equation (10) can be described as follows using the properties of bilinear mapping:

$$\begin{aligned} C'_0 &= \frac{\prod_{j\in R}(e(w,g)^{(r_x+r'_x)v_j})^{\omega_j} e(F_2(tm), g)^{(s_x+s'_x)s}}{e(g, pk_{id}^{\alpha})^s e(g,w)^{(r_x+r'_x)s} e(g, F_2(tm)^{(s_x+s'_x)s})} \\ &= \frac{e(w,g)^{(r_x+r'_x)\sum_{j\in R}\omega_j v_j} e(F_2(tm), g)^{(s_x+s'_x)s}}{e(g, pk_{id}^{\alpha})^s e(g,w)^{(r_x+r'_x)s} e(g, F_2(tm)^{(s_x+s'_x)s})} \\ &= \frac{1}{e(g, pk_{id}^{\alpha})^s}. \end{aligned} \tag{11}$$

Thus, $SymKey$ can be obtained, where

$$\begin{aligned} Symkey &= [\frac{1}{e(g,(g^{\beta})^{\alpha})^s}]^{1/\beta} e(g,g)^{\alpha s} Symkey \\ &= \frac{e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} Symkey. \end{aligned} \tag{12}$$

Then, the doctor decrypts the $CT_{HD}$ as follows:

$$HD = Decrypt(Symkey, CT_{HD}). \tag{13}$$

*Theorem 6.2:* The match can be completely correct as long as the doctor's interest set is a subset of the keyword set.

*Proof:* It is clear that every element in $\{H(inst_{t_2})\}_{t_2\in[1,m_2]}$ is the root of $P(x) = 1$. Therefore, the formula can be deduced as

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3203577

follows:

$$\begin{aligned}\prod_{t_1=0}^{m_1} E_{t_1}^{T_3 T_{5,t_1}} &= e(g,f)^{b_2 m_2^{-1} \sum_{t_1=0}^{m_1} p_{t_1} \sum_{t_2=1}^{m_2} H(inst_{t_2})^{t_1}} \cdot \\ &e(g,g)^{\alpha b_1 b_2 m_2^{-1} \sum_{t_1=0}^{m_1} p_{t_1} \sum_{t_2=1}^{m_2} H(inst_{t_2})^{t_1}} \\ &= e(g,f)^{m_2^{-1} b_2 \sum_{t_2=1}^{m_2} 1} e(g,g)^{\alpha b_1 b_2 m_2^{-1} \sum_{t_2=1}^{m_2} 1} \\ &= e(g,f)^{b_2} e(g,g)^{\alpha b_1 b_2} \\ &= e(g,f)^{b_2} e(g^{\frac{\alpha b_2}{\beta + r}}, (g^{b_1})^{\beta + r}) \\ &= T_4 e(T_1, I_1^{T_2})\end{aligned} \tag{14}$$

Thus, we have (9) to complete matching.

## B. Security Analysis

We briefly provide the security of our joint medical consultation system in the following theorem.

*Theorem 6.3:* The $Encrypt$ algorithm in Section V-B 3) is IND-CPA secure.

*Proof:* This part of the proof is shown in [24].

In addition, we mainly consider attacks executed by the following adversaries.

1. Blockchain nodes are honest but curious about the information stored in the blockchain.

2. Malicious hospitals may want to obtain patient data for medical research, leading to the disclosure of the user's privacy, or may collude with doctors to match unqualified doctors for patients for their own interests.

3. Malicious patients may modify the diagnosis results to frame the doctor for misdiagnosis.

4. A malicious doctor may deliberately make a wrong diagnosis or refuse to provide a diagnosis to the patient. However, we think that this issue is a hospital management problem and is thus beyond the scope of our research.

*Theorem 6.4:* In our design, for any probabilistic polynomial time adversaries, the attack conducted by the following adversaries is negligible if the used zk-SNARK used is of zero knowledge.

1. Curious blockchain nodes can learn information about patient health data from transactions when the data are well stored.

2. A malicious hospital may obtain the data and match incorrectly, which means the doctor's interest and index are inconsistent.

3. Malicious patients can deliberately decrypt the health report in error when the diagnosis is completed correctly.

*Proof:*

1. Patient health data are encrypted and stored on the blockchain using attribute-based encryption. The blockchain nodes have no corresponding decryption key and can rely only on a brute force attack to obtain information from the ciphertext.

2. Two steps are needed to obtain the plaintext. One is pre-decryption completed by the hospital, and the other is decryption completed by the doctor. The hospital uses the predecryption key $prek$ distributed by the KGC to complete the predecryption to obtain $SemiCT$, but the plaintext after complete decryption cannot be obtained because the hospital has no corresponding private key $sk$ that is owned by the doctor.

When completing the matching, the hospital must provide zero-knowledge proof to prove to the patient that the match is indeed completed among the doctors who meet the requirements of the patient. If the hospital's intentions are malicious, the hospital will need to forge a valid proof. However, the soundness of zk-SNARK guarantees that if the prover does not compute the proof correctly, no verifier believes the validity of the proof because of the existence of the $Verifier$ algorithm.

3. If the patient wants to decrypt $CT_{HR}$ with a tampered key to frame the doctor, it will be impossible because the valid proof includes the validity of public and private keys. The key-pairing check can prevent malicious patients from framing the doctor.

For the same reason as 2, the patient cannot forge the valid proof.

# VII. Performance Analysis

In this part, we first focus on theoretical analysis by comparing our proposed scheme with some similar schemes and show the computation costs and storage costs in Table III and Table IV. Then we evaluate the performance of our proposed design through experiments of a prototype implementation and the evaluation of its local and on-chain performance.

## A. Theoretical Analysis

Compared with prior schemes(i.e., LSABE [28], LSABE-MA [28], MABKS [49]), we show the theoretical analysis of our scheme.

In the evaluation of computational cost, we consider all the essential operations, including bilinear pairing $P$, exponentiation $E$ in group $G_1$ and $E_T$ in group $G_T$, and symmetric encryption $C_M$. $|R|$ denotes the length of set $R$. In addition to the encryption of plaintext, MABKS and our scheme also encrypt the symmetric key, so the encryption cost may be higher than LSABE and LSABE-MA. This has little impact on the search and match time of the data user because the encryption occurs on the data owner side and is a one-time cost. As for index generation and match, both MABKS and our scheme perform better than LSABE and LSABE-MA. It seems to sacrifice some time in exchange for many-to-many matching in our scheme. In fact, the index generation and match are done at negligible cost by hospitals with strong computing capability. During the decryption phase, the user only needs to perform the exponentiation operation once. This reduces the computational pressure on doctors and is appropriate for scenarios where doctors diagnose a large number of patients each day.

In addition, we compare our scheme with these similar schemes from the perspective of storage cost. Let $|Z_p|$, $|G_1|$, and $|G_T|$ denote size of elements in group $Z_p$, $G_1$, and $G_T$ respectively, $C_M$ denote the length of ciphertext of $M$, and $|A_{[j,id]}|$ denote the number of attributes obtained by the user $id$ from authority $A_j$ in LSABE and LSABE-MA. The size of decryption key in our scheme is $|Z_p|$, which is the same level as LSABE and LSABE-MA. The storage cost of match key in our scheme is much less than those of other schemes. The storage cost of ciphertext in MABKS and our scheme seems to be much higher than LSABE and LSABE-MA, this is because we perform two encryption operations. When it comes to the storage cost of index, our scheme has a slight advantage over other schemes. This means our scheme is suitable for medical scenarios, where a large amount of health data needs to be stored. Moreover, we can know from Table IV that LSABE and our scheme have the same storage cost in trapdoor, which is less than that of LSABE-MA. Although it requires only $2|G_1|$ to store the trapdoor in MABKS, it allows only a single keyword match at a time while our scheme supports many-to-many matching.

## B. Experimental Analysis

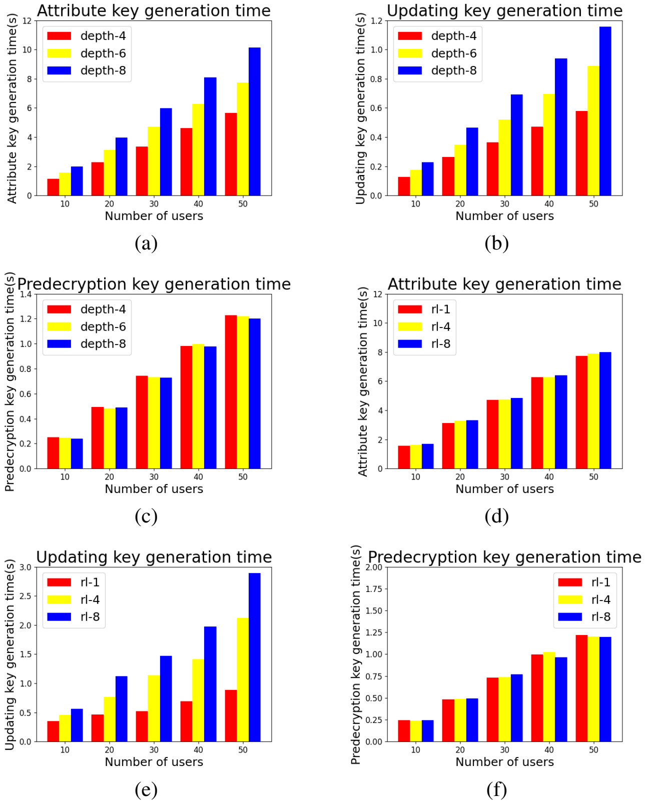
To access the performance of the scheme we designed, we implemented the prototype design in Python and built a smart contract on Ethereum with Solidity using approximately 2000 lines of code. We simulate in an Ubuntu 20.04 desktop system with Intel Core i7 and

TABLE III
COMPUTATIONAL COST COMPARISON

| Scheme | [28]-LSABE | [28]-LSABE-MA | [49]-MABKS | Ours |
|---|---|---|---|---|
| Encryption | $C_M$ | $C_M$ | $(4r+3)E+2E_T+P+C_M$ | $(4r+2)E+E_T+P+C_M$ |
| Index Generation | $(2c+4)E+3E_T$ | $(2c+4)E+3E_T+(2c+1)P$ | $(r+2)E+E_T+P$ | $2E+2m_1E_T+2m_1P$ |
| Match | $(2c+2)E+2E_T$ | $(2c+1)E+3E_T+3P$ | $2P$ | $m_1E_T+E+P$ |
| Decryption | $E_T$ | $2E_T$ | $\|R\|E_T+(2\|R\|+1)P$ | $E_T$ |

TABLE IV
STORAGE COST COMPARISON

| Scheme | [28]-LSABE | [28]-LSABE-MA | [49]-MABKS | Ours |
|---|---|---|---|---|
| Decryption Key | $\|Z_p\|$ | $\|Z_p\|$ | $(2\|R\|+2)\|G_1\|$ | $\|Z_p\|$ |
| Match Key | $\|Z_p\|+(\|A_{id}\|+3)\|G_1\|$ | $3\|A_{[j,id]}\|\|G_1\|$ | $(\|R\|+2)\|G_1\|$ | $\|Z_p\|$ |
| Ciphertext | $\|C_M\|$ | $\|C_M\|$ | $(2r+1)\|G_1\|+\|G_T\|+\|C_M\|$ | $(3r+2)\|G_1\|+\|G_T\|+\|C_M\|$ |
| Index | $(c+m_1+4)\|G_1\|+2\|G_T\|$ | $m_1\|Z_p\|+(c+4)\|G_1\|+(2c+1)\|G_T\|$ | $(r+2)\|G_1\|+\|G_T\|$ | $2\|G_1\|+m_1\|G_T\|$ |
| Trapdoor | $2\|Z_p\|+(m_1+1)\|G_1\|+\|G_T\|$ | $\|Z_p\|+(\|A_{[j,id]}+m_1+1\|)\|G_1\|+\|G_T\|$ | $2\|G_1\|$ | $2\|Z_p\|+(m_1+1)\|G_1\|+\|G_T\|$ |



(a) (b) (c) (d) (e) (f)

Fig. 3. Execution time under varying depths of the binary tree and varying revocation list lengths. (a) Attribute key generation time under varying depths. (b) Updating key generation time under varying depths. (c) Predecryption key generation time under varying depths. (d) Attribute key generation time under varying revocation list lengths. (e) Updating key generation time under varying revocation list lengths. (f) Predecryption key generation time under varying revocation list lengths.

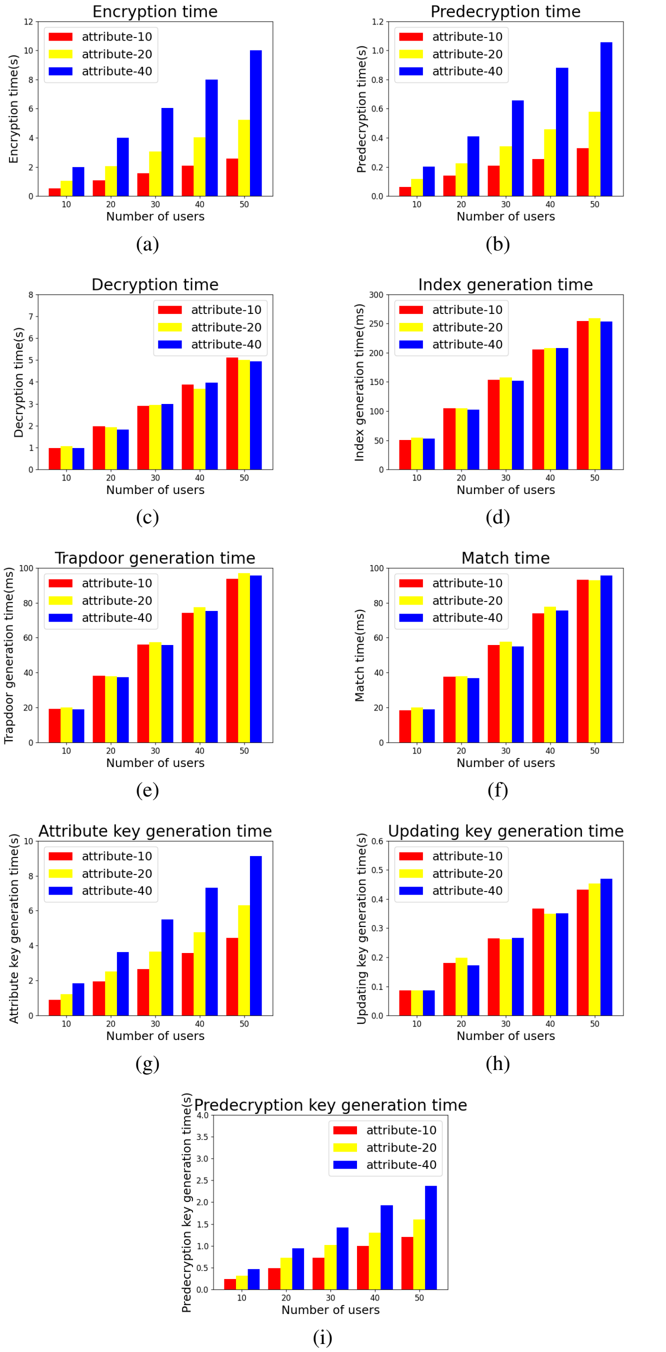
4-GB RAM. We use charm-crypto-0.5 to implement the encryption algorithm.

We evaluate the performance of the scheme from multiple perspectives including encryption, predecryption, decryption, index generation, trapdoor generation, match, attribute key generation, updating key generation, and predecryption key. The index generation time here refers to the off-chain index building time.

First, as shown in Fig. 3, we evaluate the predecryption key generation performance of the scheme. We analyze the effects of the depth of the binary tree and the number of revoked users on the generation time of the attribute key $attk$, updating key $updk$, and predecryption key $prek$. From Fig. 3(a), we know that the attribute key generation time varies as the depth of the binary tree deepens. The greater the depth of the tree is, the more users the tree can carry and the more time it takes to generate the attribute key. The change of updating key generation time is similar to the attribute key as shown in Fig. 3(b). Different from Fig. 3(a) and Fig. 3(b), in Fig. 3(c), the depth of the binary tree has no significant effect on predecryption key generation time. We agree that the best explanation for this result is that in the predecryption key generation stage, we select the elements in the intersection of sets $Path(\theta)$ and $KUNodes(BT, rl, tm)$ for operation. Although the number of elements in set $I$ and set $J$ is affected by the depth of the tree, the time taken to obtain the intersection of the two sets is negligible.

The impact of the length of the revocation list on the execution time is described below. As shown in Fig. 3(d), when the length of the revocation list is 1, 4, and 8, the generation time of the attribute key will hardly change whether it is a small number of users or a large number of users. However, things have changed for updating key. We can see from Fig. 3(e) that the length of the revocation list is 8, which means that there 8 doctors are revoked. At this time, the updating key generation can reach 2.89 s when the number of users is 50. When the length of the revocation list is 1, the time is only 0.88 s. For the predcryption key generation time, it seems that it is not easily affected by the length of the revocation list. The reason is the same as the reason why it is hardly affected by the depth of the binary tree.

Since the main encryption algorithm we adopted is attribute-based encryption, we also evaluated the impact of the number of attributes on each stage of our scheme, as shown in Fig. 4. Encryption time increases linearly with the number of attributes from 10, 20 to 40 as shown in Fig. 4(a). When the number of attributes reaches 40 and the number of patients reaches 50, the encryption time required is only approximately 10 s. For medical scenarios, the number of user attributes may be only approximately 10. Therefore, the scheme we provide is suitable for these scenarios. In the same case, the predecryption time is only approximately 1 s, so the hospital can bear the computational pressure of predecryption. The change of predecryption time is similar to that of encryption time when the number of attributes changes, that is, when the number of attributes increases, the predecryption time increases. This is because the input of predecryption is the ciphertext and the trapdoor, whose length is related to the number of attributes. For decryption executed by the doctor, it can be completed only taking $SemiCT$ and $sk$ as inputs instead of attributes. Therefore, the change in the number of



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 4. Execution time of each phase under different numbers of attributes. (a) Encryption time. (b) Predecryption time. (c) Decryption time. (d) Index generation time. (e) Trapdoor generation time. (f) Match time. (g) Attribute key generation time. (h) Updating key generation time. (i) Predecryption key generation time.

attributes has no effect on the decryption time. As shown in Fig. 4(e), we know that when the number of attributes is 40, it only takes 4.93 ms to decrypt data for which the number is 50. Experiments show that predecryption reduces doctors' computing pressure and facilitates doctors' use of computing power in other effective places. From Fig. 4(d), (e), and (f), we can see that the index generation time, trapdoor generation time, and match time do not vary significantly as the number of attributes changes. The attribute key generation time is affected by the number of attributes, while the updating key generation is not. This effect occurs because the number of $P_{x,1,i}$ and $P_{x,2,i}$ in the attribute key is equal to the number of attributes, so the size and time of the generated attribute key will be affected by attributes. For the same reason, the existence of $tk_{1,i}$ and $tk_{2,i}$ in $prek$ causes the key generation time to be related to the number of attributes.

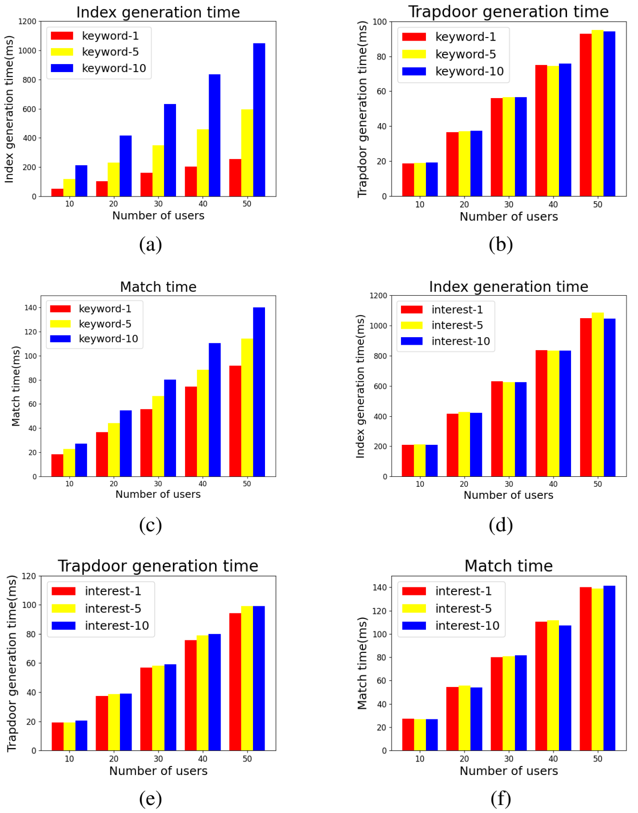
To explore the influencing factors of index generation time, trapdoor generation time, and match time, we also analyzed the changes in the three under the number of different keywords and interests, as shown in Fig. 5. The number of keywords here refers to how many keywords are selected from the patient's health data to generate an index. The index generation time is positively correlated with the number of keywords. However, the generation time of the trapdoor is not significantly affected by the number of keywords even if the generation needs to take the number of keywords as the input. The match time is affected by the number of keywords because the number of $E_{t_1}$ in $Index$ and the number of $T_{5,t_1}$ in $Trapdoor$ are related to the number of keywords. The greater the number is, the more time it takes to multiply.

We can know the relationship between time and number of interests from Fig. 5(d), (e), and (f). Since the generation of the index is unrelated to the doctor's interest, its time is also not necessarily related to the doctor's interest. As shown in Fig. 5(e), the generation of the trapdoor needs to take the doctor's interest as the input and find a polynomial that meets the conditions. The greater the number of interests is, the more time it takes to find the polynomials. However, with the increase in the number of interests, the time spent on trapdoor generation does not change significantly because the calculation time of bilinear pairs is so large that the time spent finding polynomials is negligible for the overall execution time. According to Fig. 5(f), the match time is independent of the number of interests because the match takes the trapdoor and index as inputs, and these two inputs are independent of the number of interests.

In the scheme we designed, the hospital needs to upload the index generated by itself to the blockchain. Authorization information between hospitals is stored on the blockchain. The hospital can submit its own search request, and the search is completed by the smart contract. Therefore, to evaluate the usability of the experiment, we further evaluated its on-chain performance, including index add, authorization and search. From Fig. 6(a), we know that the greater the number of indexes is, the longer the index that will be added. The time of authorization increases rapidly as the number of hospitals increases, as shown in Fig. 6(b) because each authorization from one hospital to each other is counted, which is bidirectional. As shown in Fig. 6(c), search time is positively correlated with the number of indexes. To further estimate the usability of the experiment, we evaluated the gas consumption on Ethereum, as shown in Fig. 6(d).

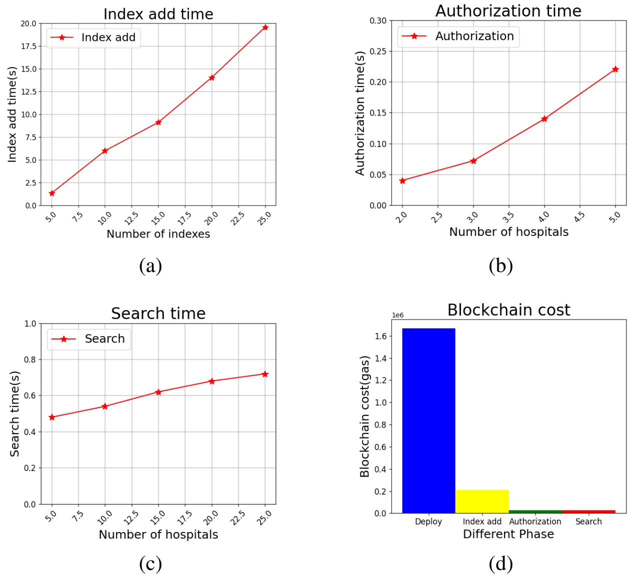
## VIII. Conclusion

We proposed a new privacy-preserving medical data-sharing scheme based on a blockchain. Our scheme realizes fine-grained access of data, many-to-many match and lightweight decryption. Specifically, we use attribute-based encryption technology to encrypt patient medical data. Through the authorization mechanism between medical institutions and publishing the authorization results to the blockchain, information sharing among multiple medical institutions can be realized. Doctors in authorized hospitals can complete diagnoses for patients in hospitals where the patient registers, and unauthorized medical institutions cannot obtain the patient's health data. Doctors can retrieve data according to their professional expertise and interests to achieve special treatment by searching executed by medical institutions. Only doctors whose attributes meet the requirements of the access policy can correctly decrypt the data



(a) (b) (c) (d) (e) (f)

Fig. 5. Execution time of index generation, trapdoor generation, and match under different numbers of keywords and interests. (a) Index generation time under different keywords. (b) Trapdoor generation time under different keywords. (c) Match time under different keywords. (d) Index generation time under different interests. (e) Trapdoor generation time under different interests. (f) Match time under different interests.



(a) (b) (c) (d)

Fig. 6. On-chain time cost and gas cost. (a) Index add time. (b) Authorization time. (c) Search time. (d) Gas cost.

to obtain plaintext for patient diagnosis, and the diagnosis results will be encrypted and returned to the patient. In addition, when the medical institution finds a match of the patient's medical data and the doctor's interests, it needs to provide zero-knowledge proof to prove that the match algorithm is indeed completed with the correct index and trapdoor to prevent the hospital from matching the patient's data at will due to its commercial interests. When the patient obtains the ciphertext of the health report, he or she needs to provide a zero-knowledge proof to prove that the decryption is indeed completed with the private key corresponding to the public key to prevent the malicious patient from framing doctors and causing medical disputes. Considering that the local computing power of doctors may be limited, we adopt the predecryption method when decrypting, and the medical institutions bear part of the computing pressure. Only one exponentiation operation and one product operation are performed to obtain plaintext. We provided correctness and security analysis and evaluated the performance of our scheme by theoretical and experimental analysis. Our analysis show that the proposed scheme is efficient, correct, well adapted in medical scenarios, and can realize medical data sharing and improve the utilization of social medical resources on the premise of protecting medical privacy.

Despite the advances of our research, several problems need to be considered in the future. The area in the joint medical consultant system remains largely unexplored. First, at present, the development of smart contracts is not mature enough, and the realization of many functions remains limited. In addition, the amount of data storage on the chain is often small. We can further optimize our scheme, realize more complex functions, and reduce the storage on the chain as much as possible to reduce the data on the chain. Second, we can use a more complete incentive mechanism to expand our implementation. We will implement the payment function in our scheme and the patient will need to pay a certain fee for the doctor's expertise. Third, some rules need to be set to standardize the whole system. Last but not least, our scheme assumes a trusted authority to complete key generation and distribution. A better solution is to develop a method that does not need a trusted third party. For example, smart contract can be constructed to replace trusted centralized KGC [50]. Another solution is that we can use the secret sharing method to reconstruct the secret from the consensus node to improve the practicability of the scheme.

## REFERENCES

[1] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks," *IEEE Trans. Netw. Sci. Eng*, vol. 8, no. 4, pp. 2946-2957, 2021.

[2] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *IEEE Trans. Netw. Sci. Eng.*, vol. 32, no. 5, pp. e5556, 2022.

[3] X. Yuan, X. Wang, C. Wang, J. Weng, and K. Ren, "Enabling secure and fast indexing for privacy-assured healthcare monitoring via compressive sensing," *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 2002-2014, 2016.

[4] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE JBHI*, vol. 18, no. 4, pp. 1431-1441, 2014.

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

[6] S. Haas, S. Wohlgemuth, I. Echizen, N. Sonehara, and G. Müller, "Aspects of privacy for electronic health records," *Int. J. Med. Informat*, vol. 80, no. 2, pp. e26-e31, 2011.

[7] Z. R. Li, E. C. Chang, K. H. Huang, and F. Lai, "A secure electronic medical record sharing mechanism in the cloud computing platform," in *Proc. 15th IEEE Int. Sympo. Consum. Electron.*, pp. 98-103, Jun. 2011.

[8] Y. Y. Chen, J. C. Lu, and J. K. Jan, "A secure EHR system based on hybrid clouds," *J. Med. Syst.*, vol. 36, no. 5, pp. 3375-3384, 2012.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE Infocom*, pp. 1-9, Mar. 2010.

[10] T. Hupperich, H. Löhr, A. R. Sadeghi, and M. Winandy, "Flexible patient-controlled security for electronic health records," in *Proc. 2nd ACM SIGHT Sympo. Int. Health Informatics*, pp. 727-732, Jan. 2012.

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3203577

[11] R. Wu, G. L. Ahn, and H. Hu, “Secure sharing of electronic health records in clouds,” in *Proc. 8th IEEE Int. Conf. Collaborative Comput., Netw., Appl. Work-sharing*, pp. 711-718, Jan. 2012.

[12] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, “Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control,” *J. Med. Syst.*, vol. 4, no. 10, pp. 1-8, 2016.

[13] B. Shen, J. Guo, and Y. Yang, “MedChain: Efficient Healthcare Data Sharing via Blockchain,” *Appl. Sci.*, vol. 9, no. 6, pp. 1207, 2019.

[14] H. L. Pham, T. H. Tran, and Y. Nakashima, “A secure remote healthcare system for hospital using blockchain smart contract,” *IEEE GC. Wkshps.*, vol. 9, no. 6, pp. 1-6, Feb. 2018.

[15] R. Zou, X. Lv, and J. Zhao, “SPChain: Blockchain-based medical data sharing and privacy-preserving eHealth system,” *IPM*, vol. 58, no. 4, pp. 102604, 2021.

[16] Z. Lian, W. Wang, H. Huang, and C. Su, “Layer-Based Communication-Efficient Federated Learning with Privacy Preservation,” *IEICE Trans Inf Syst*, vol. 105, no. 2, pp. 256-263, 2022.

[17] J. Song, W. Wang, T. R. Gadekallu, J. Cao, and Y. Liu, “Eppda: An efficient privacy-preserving data aggregation federated learning scheme,” *IEEE Trans. Netw. Sci. Eng.*, 2022.

[18] Z. Lian, Q. Yang, W. Wang, Q. Zeng, M. Alazab, H. Zhao, and C. Su, “DEEP-FEL: Decentralized, Efficient and Privacy-Enhanced Federated Edge Learning for Healthcare Cyber Physical Systems,” *IEEE Trans. Netw. Sci. Eng.*, 2022.

[19] Y. Jin, C. Tian, H. He, and F. Wang, “A secure and lightweight data access control scheme for mobile cloud computing,” in *Proc. 5th Int. Conf. Big Data Cloud Comput.*, vol. 15, pp. 172-179, Aug. 2015.

[20] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, “Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,” *EUROCRYPT*, pp. 62-91, 2010.

[21] C. Wang, W. Li, Y. Li, X. Xu, “A ciphertext-policy attribute-based encryption scheme supporting keyword search function,” *CSS*, pp. 377-386, 2013.

[22] P. Chaudhari and M. L. Das, “Privacy preserving searchable encryption with fine-grained access control,” *IEEE TCC*, vol. 9, no. 2, pp. 753-762, 2019.

[23] F. Han, J. Qin, H. Zhao, and J. Hu, “A general transformation from KP-ABE to searchable encryption,” *FGCS*, vol. 30, pp. 107-115, 2014.

[24] H. Cui, R. H. Deng, Y. Li, and B. Qin, “Server-aided revocable attribute-based encryption,” *ESORICS*, pp. 570-587, 2016.

[25] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, “Blockchain empowered arbitrable data auditing scheme for network storage as a service,” *IEEE TSC*, vol. 13, no. 2, pp. 289-300, 2019.

[26] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, “Blockchain-enabled accountability mechanism against information leakage in vertical industry services,” *IEEE Trans. Netw. Sci. Eng*, vol. 18, no. 2, pp. 1202-1213, 2020.

[27] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, “A blockchain-enabled deduplicatable data auditing mechanism for network storage services,” *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1421-1432, 2020.

[28] K. Zhang, J. Long, X. Wang, H. Dai, K. Liang, and M. Imran, “Lightweight searchable encryption protocol for industrial internet of things,” *IEEE Trans. Industr. Inform.*, vol. 17, no. 6, pp. 4248-4259, 2020.

[29] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu, and R. Y. K. Kwork, “Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework,” *TMC*, pp. 1-1, 2021.

[30] B. Wang, W. Song, W. Lou, and Y. T. Hou, “Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee,” in *Proc. IEEE INFOCOM*, pp. 2092-2100, 2015.

[31] Z. Guan, G. Si, X. Zhang, L. Wu, N. Gruizani, X. Du, and Y. Ma, “Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities,” in *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 82-88, Jul. 2018.

[32] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *Proc. IEEE Int. Congr. Big Data, Big Data Congr.*, pp. 557-564, Jun. 2017.

[33] R. Mishra, D. Ramesh, D. R. Edla, X. Liu, R. Lu, H Li, and Y. Zhang, “Deletable Blockchain based Secure EHR Storage Scheme in Multi-Cloud Environment,” *IEEE DSS*, pp. 1057-1064, 2020.

[34] V. Casola, A. Castiglione, K. R. Choo, and C. Esposito, “Healthcare-related data in the cloud: Challenges and opportunities,” *IEEE Cloud Comput.* , vol. 3, no. 6, pp. 10-14, 2016.

[35] K. Liang and W. Susilo, “Searchable attribute-based mechanism with efficient data sharing for secure cloud storage,” *IEEE Trans. Inf. Forensics Security.*, vol. 10, no. 9, pp. 1981-1992, 2015.

[36] H. Lin, J. Shao, C. Zhang, and Y. Fang, “CAM: cloud-assisted privacy preserving mobile health monitoring,” *IEEE Trans. Inf. Forensics Security.*, vol. 8, no. 6, pp. 985-997, 2013.

[37] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” *S&P*, pp. 839-858, 2016.

[38] R. A. Popa and N. Zeldovich, “Multi-key searchable encryption,” *Cryptol. ePrint Archive*, vol. 2013, pp. 508, 2013.

[39] D. Koo, J. Hur, and H Yoon, “Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage,” *Comput. Elect. Eng.*, vol. 39, no. 1, pp. 34-46, 2013.

[40] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, “Flexible and fine-grained attribute-based data storage in cloud computing,” *IEEE TSC*, vol. 10, no. 5, pp. 785-796, 2016.

[41] J. Wei, W. Liu, and X. Hu, “Secure and efficient attribute-based access control for multiauthority cloud storage,” *IEEE Syst. J*, vol. 12, no. 2, pp. 1731-1742, 2016.

[42] K. Sowjanya, M. Dasgupta, S. Ray, and M. S. Obaidat, “An efficient elliptic curve cryptography-based without pairing KPABE for internet of things,” *IEEE Syst. J*, vol. 14, no. 2, pp. 2154-2163, 2019.

[43] K. Sowjanya, M. Dasgupta, S. Ray, and M. S. Obaidat, “AKSER: Attribute-based keyword search with efficient revocation in cloud computing,” *Inf. Sci.*, vol. 423, pp. 343-352, 2018.

[44] J. Bethencourt, A. Sahai, and Brent Waters, “Ciphertext-policy attribute-based encryption,” *S&P*, pp. 321-334, 2007.

[45] P. Tasatanattakool and C. Techapanupreeda, “Blockchain: Challenges and applications,” *ICOIN*, pp. 473-475, 2018.

[46] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[47] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, “Succinct {Non-Interactive} Zero Knowledge for a von Neumann Architecture,” *Security*, pp. 781-796, 2014.

[48] H. Cui, R. H. Deng, J. K. Liu, and Y. Li, “Attribute-based encryption with expressive and authorized keyword search,” *ACISP*, pp. 106-126, 2017.

[49] Y. Miao, R. H. Deng, X. Liu, K. R. Choo, H. Wu, and H. Li, “Multi-authority attribute-based keyword search over encrypted cloud data,” *TDSC*, vol. 18, no. 4, pp. 1667-1680, 2019.

[50] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, “Blockchain-based reliable and efficient certificateless signature for IIoT devices,” *IEEE Trans. Ind. Inf.*, pp. 1-9, May. 2021.

[51] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, “FPDP: flexible privacy-preserving data publishing scheme for smart agriculture,” *IEEE Sens. J*, vol. 21, no. 16, pp. 17430-17438, Aug. 2020.

[52] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, “Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks,” *IEEE Internt Things J.*, vol. 9, pp. 8883-8891, Oct. 2021.

[53] B. Aslam, A. R. Javed, C. Chakraborty, J. Nebhen, and S. Raqib, “Blockchain and ANFIS empowered IoMT application for privacy preserved contact tracing in COVID-19 pandemic,” *PUC*, pp. 1-17, 2021.

[54] T. F. Lee, H. Z. Li, and Y. P. Hsieh, “A blockchain-based medical data preservation scheme for telecare medical information systems,” *IJISP*, vol. 20, no. 4, pp. 589-601, 2021.

[55] R. Ch, G. Srivastava, T. R. Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, “Security and privacy of UAV data using blockchain technology,” *JISA*, vol. 55, pp. 102670, 2020.

[56] T. R. Gadekallu, Q. V. Pham, D. C. Nguyen, P. K. R. Maddikunta, N. Deepa, B. Prabadevi, P. N. Pathiranan, J. Zhao, and W. J. Hwang, “Blockchain for edge of things: applications, opportunities, and challenges,” *IEEE Internt Things J.*, vol. 9, no. 2, pp. 964-988, 2021.

[57] G. Xu, W. Dong, J. Xing, W. Lei, J. Liu, L. Gong, M. Feng, X. Zheng, and S. Liu, “Delay-CJ: A novel cryptojacking covert attack method based on delayed strategy and its detection,” *Digit. Commun. Netw.*, 2022.

[58] G. Xu, H. Bai, J. Xing, T. Luo, N. N. Xiong, X. Cheng, S. Liu, and X. Zheng, “SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles,” *JPDC*, vol. 164, pp. 1-11, 2022.