IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 April 2024, accepted 5 May 2024, date of publication 13 May 2024, date of current version 21 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400611

RESEARCH ARTICLE

An Inner Product Predicate-Based Medical Data-Sharing and Privacy Protection System

ZIGANG WU, HAIJIANG WANG, JIAN WAN, LEI ZHANG, AND JIE HUANG
School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Corresponding author: Haijiang Wang (wanghaijiangyes@163.com)

This work was supported in part by the Zhejiang Provincial Key Research and Development Program Project under Grant 2021C03145; in part by the National Natural Science Foundation of China under Grant 61972358; and in part by the Natural Science Foundation of Zhejiang Province, China under Grant LQ20F020010.

ABSTRACT Searchable encryption is widely used in electronic health record systems because it enables users to search ciphertext data without decryption. However, the existing traditional searchable encryption schemes lack fine-grained access policies with wildcards in electronic health record systems. And they also do not consider the problem of hiding policies, as well as the problem of incomplete search results caused by cloud servers. In order to solve all the above problems, this paper proposes a blockchain-aided attribute-based searchable scheme with the properties of inner product predicate. In the proposed scheme, the attribute encryption mechanism implements fine-grained access policies with wildcards, which improves the data owner's ability to control access authorization precisely. Introducing the inner product predicate not only achieves fully hidden access policies but also prevents the leakage of sensitive medical data. The immutability of the blockchain ensures the integrity of multi-keyword search results, guaranteeing reliable data sharing. Finally, the security proof and the performance evaluation are conducted to confirm the security and effectiveness of the proposed scheme.

INDEX TERMS Attribute-based searchable encryption, fine-grained access policies with wildcards, fully hidden, inner-product predicate, blockchain.

I. INTRODUCTION

A. BACKGROUND

Cloud computing [1], as a new technology that provides computing and storage capabilities. Unlike traditional local data storage mechanisms, cloud computing utilizes various networks to reconstruct vast computing and storage resources into an efficient logical entity, providing convenient storage services and personalized computing. Its efficiency, precision and convenience have granted cloud computing a prominent role in smart healthcare. However, the cloud computing storage model faces significant security risks in the form of malicious attackers and network viruses. Data owners lose control of their data when hospitals upload electronic health records (EHRs) to cloud servers, meaning that patients' privacy [2] may be compromised if malicious attackers successfully access the stored data. Currently, encryption technology is the tool most frequently used to prevent such privacy breaches, allowing data owners to encrypt their information and upload it to cloud servers for storage. This prevents malicious attackers from accessing plaintext information and effectively protecting people's privacy.

Encryption methods are generally divided into two types: symmetric encryption and asymmetric encryption. Symmetric encryption is advantageous due to its rapid encryption and decryption speed; however, its primary disadvantage lies in the complexity of distributing and managing keys. The introduction of asymmetric encryption aims to address these intricacies. However, traditional asymmetric encryption mechanisms not only require that authorized institutions issue public key certificates related to the user's identity, but also require that the corresponding public key certificates be queried from authorized institutions before the encryption process can take place, which negatively affects the authority's service performance. The emergence of identity-based encryption [3] solves the time-consuming problem of public

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

key certificate queries. Regrettably, however, identity-based encryption still has its limitations. Specifically, the algorithm only supports one-to-one encryption scenarios and is not suitable for one-to-many applications.

Hence, attribute-based encryption (ABE) [4], including ciphertext policy attribute-based encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE), is used to realize data sharing in one-to-many scenarios. In the CP-ABE encryption mechanism, a policy is embedded in the ciphertext, whereas in the KP-ABE encryption mechanism, a policy is embedded in the key. This not only enables fine-grained one-to-many access control but also guarantees the security of ciphertext data in ABE.

However, data users cannot effectively search ciphertext data once the encryption is complete. Therefore, searchable encryption (SE) [5], [6], [7], [8] as a cryptographic primitive addresses the scenario in which data users perform searchable operations on ciphertext data during the decryption process. Symmetric searchable encryption (SSE) and public-key searchable encryption (PKSE) are two different categories of SE. Since the encryption mode of SSE is similar to symmetric encryption, it is also suitable for one-to-one searches. Meanwhile, the PEKS encryption mode is suitable for one-to-many searches due to its similarity to asymmetric encryption. It is clear, therefore, that the user's search expression capability is one of the most important features to consider when designing SE schemes.

Currently, PEKS is commonly classified into single-keyword searchable encryption and multi-keyword searchable encryption, based on the different expressive capabilities of users' chosen keywords. Single-keyword searchable encryption only allows users to utilize a single keyword in generating the trapdoor, which can lead to the presence of a great deal of irrelevant information in the search results. With multi-keyword searchable encryption addressing the redundancy issue in the search space of single-keyword searchable encryption, multiple keywords can be used to generate the search trapdoor. This significantly improves the accuracy of users' searches.

Although existing SE mechanisms provide greater search functionality in relation to encrypted data, they lack fine-grained access policies. In order to achieve both fine-grained access policies and searchable functionality over encrypted data, many attribute-based searchable encryption (ABSE) [9], [10], [11], [12] schemes combine the functional features of ABE and SE to realize fine-grained access policies with wildcards and ciphertext searchability. To make such mechanisms more suitable for cloud environments, we have designed a more secure and feature-rich attribute-based searchable encryption scheme for the protection and sharing of medical data, which has high theoretical and practical value.

B. RELATED WORK

The ABSE mechanism prevents data privacy leakage during medical data sharing due to its ability to meet complex data-sharing requirements. However, it also shares the same shortcomings as the ABE mechanism since the ABSE mechanism inherits its characteristics. In other words, it lacks sufficient security protection for the policies embedded in the ciphertext. Typically, access policies contain sensitive medical information, meaning that if a malicious adversary is able to infiltrate them, they can also access patients' medical data. Of course, this is unacceptable.

The forms of policy protection can be divided into two categories in the ABSE mechanism: partially hidden policies and fully hidden policies.

Partially hidden access policies are those in which the attribute values are hidden but the attribute names are public. Many ABSE schemes [13], [14], [15], [16], [17] that support partially hidden AND-gates on multi-valued access policies have been proposed. In order to construct AND-gates on multi-valued access policies, data owners not only need to split attribute names and attribute values but must also select random numbers to partially sighted the attribute values. Even though supporting AND-gates on multi-valued access policies offers considerable flexibility, there is still room for further expansion. To continue to enhance the flexibility of access policies, Wu et al. [18] proposed an ABSE scheme supporting partially hidden linear secret sharing (LSS) access policies. An LSS access policy reorganizes secrets using secret linear reconstruction attributes, enabling data owners to formulate arbitrary conjunctive and disjunctive threshold authorization controls. In addition, Yan et al. [19] proposed an ABSE scheme that supports partially hidden tree access policies. A tree policy also incorporates disjunctive and conjunctive access control. However, a drawback of such policies is that the system's performance encounters bottlenecks as the depth of recursion increases. More importantly, the aforementioned schemes only support partial hiding of access policies, meaning that the attribute names remain public.

To achieve full hiding of access policies, in which both attribute names and attribute values are anonymized, one feasible approach is to introduce predicate encryption (PE) [20]. More specifically, PE, as a new paradigm of encryption mechanisms allows data owners to establish an access policy regarding how users can decrypt data without being able to view any of the relevant information. If user attributes satisfy the ciphertext access policy, in other words, if the evaluation of the inner product predicate is 1, decryption will be fulfilled. Therefore, achieving fully hidden policies requires combining ABSE with the concept of the inner product predicate [21]. The evaluation of inner product predicate will be 1 if a dot product operation is equal to 0; otherwise, it will be 0.

In 2021, Meng et al. [22] designed two access policies embedded in the ciphertext in the ABSE scheme: one is public, while the other is sensitive and hidden. If a user's attributes do not satisfy the public policy, they cannot access any information (including attribute names and their values) of the hidden policy. However, this scheme cannot guarantee that all access policies will be fully hidden. Subsequently, Najafi et al. [23] proposed an ABSE scheme that facilitates a fully hidden policy by combining Vietès formulas [24] with inner product predicate. Regrettably, however, the access policy expression capability of their scheme is limited to AND-gates on +/− with wildcards. Compared with access policies supporting AND-gates on +/− with wildcards, access policies that support AND-gates on multi-valued attributes with wildcards are more expressive. Therefore, the next step is to enhance the expressiveness of fully hidden policies in ABSE schemes.

Additionally, the ABSE mechanism shares the same drawbacks as the SE mechanism since it evolves from SE. More specifically, the ABSE mechanism lacks a more precise keyword search function.

To enable the sharing and exchange of encrypted EHRs among different hospitals and medical institutions, Guo et al. [25] proposed an ABSE scheme that enables decryption without a secure channel. This not only supports searchable functionality on encrypted EHRs data, but also enables decryption operations on the encrypted EHRs data. However, it does not support effective hiding of access control policies. In order to combat this, Zhang et al. [26] proposed a powerful data protection mechanism within the ABSE scheme which allows data users to search encrypted EHRs while partially hidden access policies. Unfortunately, this scheme cannot verify the authenticity of data search results. In their work, Liu et al. [27] proposed an ABSE scheme that supports signature verification for EHRs, implementing fine-grained access policies, data validation, and data searching. Unfortunately, the above-mentioned ABSE schemes only support single-keyword search operations, meaning that data users, such as medical institutions, are unable to precisely search through electronic health record (EHR) files using multiple keywords.

In view of the above deficiencies, Yin et al. [28] and Wu et al. [29], respectively, proposed ABSE schemes that allow multi-keyword searches to be performed in EHR systems. Their schemes enable users to perform search operations on encrypted EHRs by combining multiple keywords. However, the drawback of these schemes is their inability to effectively resist keyword-guessing attacks. To this end, Miao et al. [30] proposed an ABSE scheme that utilizes a random oracle to resist keyword-guessing attacks during the security proof process. Furthermore, to enhance the search pattern with multiple keywords, He et al. [31] created an ABSE scheme that supports Boolean keyword searches. Keywords can be divided into keyword names and keyword values, allowing users to search according to their preferences and reducing the risk of keyword leakage. Recently, Chen et al. [32] also constructed an ABSE scheme using a dual-server model to support multi-keyword search ranking. Their scheme incorporates two servers, which complete the search process cooperatively and restrictively. However, the exposed access policy is susceptible to attacks from external adversaries. To address this issue, Sun and Xu [33] proposed a multi-keyword search ABSE scheme that supports partially hidden policies. However, despite the policy being in a semi-hidden state on the server, there is still a risk of leakage.

Cloud servers can provide convenient and massive medical data storage services for hospitals. However, they also present significant security issues [34]. If unauthorized users can access cloud servers arbitrarily, private medical data will be leaked, affecting the hospitals' credibility. Fortunately, development and application of blockchain technology [35], [36] have provided new opportunities to address such issues.

To prevent malicious users and cloud service providers performing unauthorized searches of encrypted EHR files, Chen et al. [37] proposed an SE scheme based on blockchain storage in EHR systems. This scheme allows data owners to limit access to their EHRs because only indexes are migrated to the blockchain to facilitate dissemination. In addition, ABE, especially encryption that embeds attributes into ciphertext, plays a significant role in medical data sharing. However, ABE does not provide the ability to decrypt the data in time for users to conduct search operations [38]. Additionally, there is a risk of data tampering by cloud servers in ABE. Blockchain technology can ensure the integrity of medical data search results. Gupta et al. [39] proposed a blockchain-assisted ABSE scheme in the personal health protection environment. In their scheme, the decentralized nature of the scheme and the absence of a trusted authority protect the integrity of search results. In the above-mentioned schemes, data users are limited to single-keyword searches, which is impractical for real-world medical data retrieval.

Niu et al. [40] proposed an ABSE scheme based on blockchain for multi-keyword searches. Data users search encrypted EHR files using multiple keywords. The presence of blockchain also ensures the reliability of multi-keyword search results. Additionally, many blockchain-based ABSE schemes [41], [42], [43], [44], [45] have been proposed to support multi-keyword search. The characteristics of blockchain are effectively utilized in ABSE schemes to protect the confidentiality of EHR keyword ciphertext.

C. OUR MOTIVATIONS

Our Motivations can be summarized as follows:

(1) Flexible access policies. Strengthening data owners' ability to develop flexible access control policies has become the focus of research in EHR systems that support ABSE. Compared to rigid access control policies, flexible access policies allow data owners such as hospitals to implement more effective search-authorized access control.

(2) Policies protection. Full anonymity of policies is required when sharing EHRs. Access policies are embedded into the encrypted EHRs within the ABSE mechanism. If a malicious adversary eavesdrops on the ciphertext data stored on cloud services, the disclosure

of access policies will expose patients' private medical information.

(3) Reliable search results. Ensuring that correct and complete search results are returned to the user is a controversial issue in ABSE-enabled EHR systems. Generally, data owners store encrypted EHR files on cloud servers, which are typically considered to be a semi-trusted entity. Therefore, there is a possibility that cloud servers may engage in malicious tampering of search results when conducting search operations. If data users are unable to obtain correct search results, the effective sharing of EHRs will be affected.

D. OUR TECHNIQUES

(1) Vietè's formulas: This is an important tool for extracting policies and user's attributes coefficients. The theorem uses the invariance of polynomial product and power term summation equations, allowing the construction of policies and user properties.

(2) Inner product predicate: This is an important property in FE, which can be combined with ABSE to fully hide access policies. More specifically, it ensures that an adversary holding tokens $tk_{f_1}, \ldots, tk_{f_h}$ for predicates $f_1, \ldots, f_h$ cannot derive any information on attribute X from ciphertext ct_X other than the values of $f_1(X), \ldots, f_h(X)$. In the above encryption mechanism, a series of predicates are called inner product predicates. More formally, the attributes of the inner product predicate are represented as vector $\overrightarrow{X}$ and predicate $f_{\overrightarrow{Y}}$ is connected with vector $\overrightarrow{Y}$, where $f_{\overrightarrow{Y}}(\overrightarrow{X}) = 1$ iff $\overrightarrow{Y} \cdot \overrightarrow{X} = 0$.

(3) Blockchain: Possessing the characteristics of decentralization and tamper-proofing, blockchain can resolve a number of security issues associated with centralized servers, such as cloud storage data leakage, information tampering, etc., without requiring the participation of a trusted third party.

E. OUR CONTRIBUTIONS

With the above motives in mind, we design a blockchain-aided attribute-based searchable scheme with the properties of inner product predicate. Our contributions can be summarized as follows:

(1) Fine-grained access policies with wildcards: The access policy implements AND-gates on multi-valued attributes with wildcards, which extends from AND-gates on $+/-$ with wildcards. Data owners can formulate more fine-grained access policies by transforming policies using the Vietè's formulas.

(2) Fully hidden policies: A fully hidden access policy is implemented by the inner product predicate to avoid data privacy disclosure.

(3) Integrity of search results: When blockchain is introduced as an index storage space, it not only ensures the accuracy and integrity of search results during multi-keyword search operations but also addresses the semi-trust issue associated with cloud services.

F. PAPER ORGANIZATION

The subsequent sections of the paper are organized as follows: Section II outlines the preliminaries. The system model, definitions, and the security model are outlined in Section III. A comprehensive blockchain-aided EHRs sharing scheme is described in Section IV. The security proof is presented in Section V. A comparative performance analysis is conducted in Section VI and the conclusions are listed in Section VII.

II. PRELIMINARIES

A. BILINEAR PAIRINGS

This article will view G and G_T as two cyclic groups with prime order p. $e : G \times G \rightarrow G_T$ possesses the following properties:

- Bilinearity: $\forall g, \varphi \in G, a, b \in \mathbb{Z}_q, e(g^a, \varphi^b) = e(g, \varphi)^{ab}$.
- Computability: $\forall g, \varphi \in G$, there is an algorithm that can efficiently compute $e(g, \varphi)$.
- Non-degenerate: $\exists g \in G, e(g, g) \neq 1$.

B. ASSUMPTION OF COMPLEXITY

The Decisional Linear (DLIN) Assumption: We choose $a, b, c, d \in \mathbb{Z}_q$ at random and providing a tuple $(g, g^a, g^b, g^{ac}, g^d, Z)$ to determine whether $Z = g^{b(c+d)}$ or $Z = g^r$. If $\mathcal{A}$ possesses a significant advantage ε for any probabilistic polynomial-time algorithm then the assumption will be upheld.

$$\begin{aligned} &Adv_{\mathcal{A}}^{DLIN}(\lambda) \\ &\quad = |Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^d, g^{b(c+d)}) = 0] \\ &\qquad - Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^d, g^r) = 0]| < \varepsilon \end{aligned}$$

C. ACCESS POLICIES

The AND-gates on multi-valued attributes with wildcards are supported in the access structure, where each attribute with multi-value. Let $U = \{Att_1, Att_2, \cdots, Att_L\}$ represent the attribute universe in the system. For each $Att_i \in U$, $V_i = \{v_1, v_2, \cdots, v_m\}$ is the set of possible values, where m is the maximum number of values for each Att_i. When a user takes part in the system, an attribute list $S' = \{S'_1, S'_2, \cdots, S'_L\}$ is determined by the authority to describe the user, where $S'_i \in V_i$. Let $\mathbb{A} = \{S_1, S_2, \cdots, S_L\}$ denote an AND-gates on multi-valued attributes with wildcard access policy, where $S_i \in V_i$. The notation $S'| = \mathbb{A}$ indicates that the user's attribute list S' satisfies the access policy $\mathbb{A}$ if and only if $S'_i = S_i$ or $S_i = *$ for all $1 \leq i \leq L$ and $S'_i| \neq S_i$; otherwise, $S'| \neq \mathbb{A}$. The wildcard '$*$' refers to 'don't care' attribute values.

According to Table 1, we assume that U is {Field, Department, Identity}. $\mathbb{A}_0$ and $\mathbb{A}_1$ are two access policies, respectively. One access policy $\mathbb{A}_0$ is {Phylaxiology, None,

TABLE 1. **Policies and user attributes.**

Attributes		Att_1	Att_2	Att_3
Attribute name		Field	Department	Identity
Attribute value	v_1	Phylaxiology	Internal medicine department	Archiater doctor
	v_2	Sanipratics	Surgery department	Associate doctor
	v_3	Rehabilitation	Pediatric department	Physician assistant
$\mathbb{A}_0$		v_1	$*$	v_3
$\mathbb{A}_1$		v_2	v_1	$*$
Medical institute A		v_1	v_3	v_3
Medical institute B		v_2	v_1	v_2

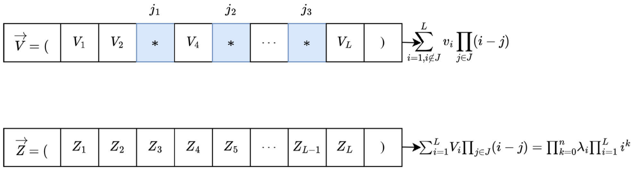


FIGURE 1. **The expression of two vectors.**

Physician assistant}; the other access policy $\mathbb{A}_1$ is {Sanipratics, Internal medicine department, None}. In addition, there are two medical institutions, which are referred to as A and B. The attributes set of medical institution A is {Phylaxiology, Pediatric department, Physician assistant}, while the attributes set of medical institution B is {Sanipratics, Internal medicine department, Associate doctor}. It can be concluded that the attributes set of medical institution A satisfies policy $\mathbb{A}_0$ and the attributes set of medical institution B satisfies policy $\mathbb{A}_1$.

D. SEARCH STRUCTURE

Let $F = \{f_1, \cdots, f_D\}$ represent the set of files of the data owner in the system. Assume that $K = \{w_1, \cdots, w_l\}$ is the possible keywords set. The data owner extracts the keyword set $W = \{w_1, \cdots, w_{l_1}\}$, where for each file f_i, $i \in [1, D]$, $l_1 \leq l$. Let $Q = \{w_1, \cdots, w_s\}$, where $s \leq l$, representing the number of search keywords. The symbol $Q| = W$ indicates that Q can be matched W if and only if $Q \subseteq W$ for each f_i, where $i \in [1, D]$.

E. VIETÈ'S FORMULAS

We observe two vectors $\overrightarrow{v} = (v_1, \cdots, v_L)$ and $\overrightarrow{z} = (z_1, \cdots, z_L)$ in Figure 1, where the vector $\overrightarrow{v}$ represents alphabets and wildcards and the other vector $\overrightarrow{z}$ represents only the location of the wildcard. $J = \{j_1, \cdots, j_n\} \subset \{1, \cdots, L\}$ represents the location of the wildcards in the vector $\overrightarrow{v}$.

Set $\prod_{j\in J}(i-j) = \sum_{k=0}^{n} \lambda_k i^k$, where λ_k are the coefficients that rely on J, and therefore

$$\sum_{i=1,i\notin J}^{L} v_i \prod_{j\in J}(i-j) = \sum_{k=0}^{n} \lambda_k \sum_{i=1}^{L} z_i i^k$$

if $v_i = z_i \vee = *$ for $i = 1, \cdots, L$.

We select C_i from the group element and let v_i, z_i act as the exponents of C_i. Then, Figure 1 becomes

$$\prod_{i=1,i\notin J}^{L} C_i^{v_i \prod_{j\in J}(i-j)} = \prod_{k=0}^{n} (\prod_{i=1}^{L} C_i^{z_i i^k})^{\lambda_k}$$

According to the Vietè's formulas, we compute the coefficient λ_k in Figure 1 as follows:

$$\lambda_{n-k} = (-1)^k \sum_{1\leq i_1 < i_2 < \cdots < i_k \leq n} j_{i_1} j_{i_2} \cdots j_{i_k}, 0 \leq k \leq n$$

where $n = |J|$.

F. BLOCKCHAIN

Blockchain: Blockchain is a trusted, decentralized, distributed database that is jointly maintained by all participants. It connects data blocks in the form of a chain in time order and ensures that the data are not tampered with, remain unforgeable, and are traceable through cryptography in the peer-to-peer (P2P) network environment.

According to the authority of node management and the scope of the network, blockchains are categorized as public, permissioned, and consortium blockchains. Public blockchains are open to anyone with an internet connection who can send a transaction to said blockchain and receive verification. Typically, such networks provide financial incentives for those who join a blockchain node and leverage some type of proof-of-stake or proof-of-work algorithm. Such incentives include bitcoin [46] and ethereum [47]. Meanwhile, permissioned blockchains cannot be joined without an invitation from the network administrator. Consortium chains are often considered to be semi-decentralized. For these blockchains, after permission has been granted, other participants may join the network and gain access rights which will allow them to control the activities of the participants.

III. SYSTEM OVERVIEW

The system overview is presented in this section.

A. SYSTEM MODEL

As shown in Figure 2, the system model consists of six parts: patient, hospital, medical institution, cloud server, blockchain, and trusted authority, as listed below:

(1) **Patient**: A patient usually requires a physical exam to supplement their treatment. The hospital then stores

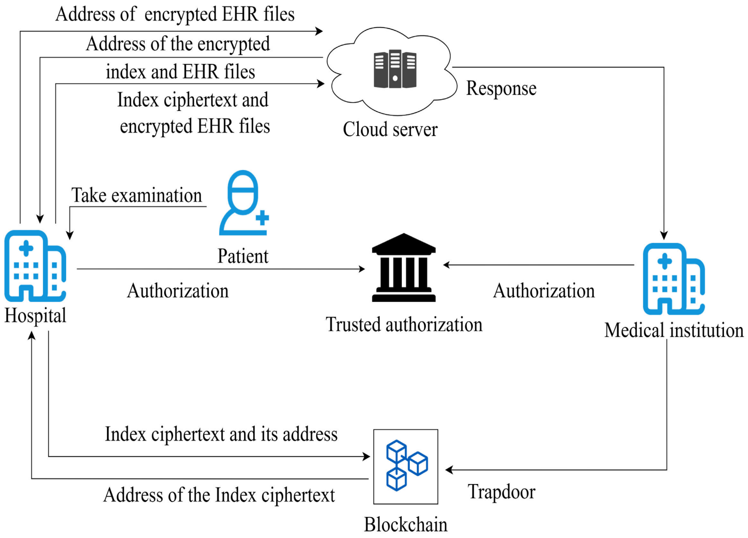

Address of encrypted EHR files
Address of the encrypted index and EHR files
Index ciphertext and encrypted EHR files
Cloud server
Response
Take examination
Patient
Hospital
Authorization
Trusted authorization
Authorization
Medical institution
Index ciphertext and its address
Address of the Index ciphertext
Blockchain
Trapdoor

FIGURE 2. The EHRs system model on the blockchain.

their test results in the form of EHRs, allowing doctors to diagnose the patient and make judgments about their condition.

(2) **Hospital**:As the data owner, the hospital has access to the paient's EHRs. The hospital uses defined access policies to encrypt the keyword collection extracted from EHR files and upload it to the cloud service storage. The cloud server will return the corresponding storage address to the hospital once it has received the ciphertext. The reverse indexing relationship between encrypted EHR files and ciphertext addresses is established in the cloud server. The hospital embeds the index ciphertext and its storage address into a transaction, which is then uploaded to blockchain storage.

(3) **Medical institution**: The data user, in other words, the medical institution wishes to conduct further research based on the patient's EHRs. Therefore, the medical institution generates a trapdoor using its own attributes and search keywords and uploads this trapdoor to the blockchain in the form of a transaction. The cloud server retrieves the encrypted EHR file based on the address and returns the ciphertext of the data file to the medical institution.

(4) **Cloud server**: As a semi-trusted entity, the cloud service provides storage and download services for encrypted EHRs. Once the cloud server receives the uploaded index ciphertext and encrypted EHR files, it will return the stored address to the hospital. The hospital uses the address returned by the blockchain to view the corresponding index relationship and sends a request to the cloud server. The cloud server then conducts a search based on the ciphertext of the data file and returns the resulting information to the medical institution.

(5) **Blockchain**: The blockchain executes the search operation. Nodes on the blockchain run search algorithms to obtain rewards based on the incentive mechanism. If the search is successful, the node on the blockchain returns the ciphertext storage address of the keyword to the medical institution; Otherwise, it returns a failure.

(6) **Trusted authorization**: The trusted authorization is responsible for creating the public key and the master secret key.

B. SCHEME DEFINITION

The system defines the algorithm as follows:

- $Setup(\lambda, U) \rightarrow (PK, MSK)$. The *Setup* algorithm takes security parameters λ and attributes universe set U as input. The public key PK and master secret key MSK are used as the output of the *Setup* algorithm.
- $IndGen(PK, W, \mathbb{A}) \rightarrow CT$. The *IndGen* algorithm inputs the public parameters PK, the extracted keywords set W, and an access policy $\mathbb{A}$. Finally, it outputs an index CT as the ciphertext of the extracted keywords set.
- $TrapGen(PK, MSK, S', Q) \rightarrow T$. The public parameters PK, the master secret key MSK, a set of user's attributes S', and the search keywords Q are inputted into the *TrapGen* algorithm to generate trapdoor T.
- $Search(CT, T) \rightarrow \{1, 0\}$. The *Search* algorithm inputs the index CT and the trapdoor T. If the search keywords match the index, the search algorithm outputs 1; otherwise, it outputs 0.

C. SECURITY MODEL

The security challenge is as listed below:

- Init. Two challenge access policies $\mathbb{A}_0, \mathbb{A}_1$ and two extracted keyword sets W_0, W_1 are submitted by the adversary $\mathcal{A}$.

- Setup. The setup algorithm is run by challenger $\mathcal{B}$ to create the public key PK and send it to the adversary $\mathcal{A}$.
- Phase 1. The adversary $\mathcal{A}$ sends the set of attributes S'' and the set of search keywords Q' to the challenger $\mathcal{B}$. If $(S''| = \mathbb{A}_0 \wedge S''| = \mathbb{A}_1)$ or $(S''| \neq \mathbb{A}_0 \wedge S''| \neq \mathbb{A}_1)$ or $(Q'| = W_0 \wedge Q'| = W_1)$ or $(Q'| \neq W_0 \wedge Q'| \neq W_1)$, the challenger $\mathcal{B}$ will generate the trapdoor T. The adversary $\mathcal{A}$ can repeat these queries in polynomial time.
- Challenge. The challenger $\mathcal{B}$ randomly flips a coin $b \in \{0, 1\}$ and sends $CT_{\mathbb{A}_b, W_b}$ to adversary $\mathcal{A}$.
- Phase 2. The query of Phase 1 will be repeated by the adversary $\mathcal{A}$.
- Guess. The adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$. Therefore, the adversary $\mathcal{A}$ wins the game on the condition that $b' = b$. The advantage of adversary $\mathcal{A}$ is as listed below:

$$Adv_A(\lambda) = |Pr[b' = b] - \frac{1}{2}| < \epsilon$$

where ϵ is negligible.

IV. SCHEME DETAILS

A. SCHEME CONSTRUCTION

Let l represent the maximum number of keywords in an index obtained from an EHR file.

(1) *Setup*(λ, U): If the universe U has L categories of attributes, and each attribute possesses m potential values, we assume that wildcards $*$ represent so-called "don't care" attributes of access policies. Set N_0, N_i includes two upper bounds defined as:

$N_0 \leq L$: the maximum number of wildcards in an access policy;

$N_i \leq L$: the maximum number of attributes with i^{th} value in a user's attribute S';

The setup algorithm arbitrarily generates (p, G, G_T, g, e) through the security parameter λ. Next, the system randomly selects $w_{1,i}, w_{2,i}, w_{3,i}, w_{4,i}, z_{1,i}, z_{2,i}, z_{3,i}, z_{4,i}$ $\gamma, \theta \leftarrow \mathbb{Z}_q$, $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$, $g_3 \in G$. Then, it randomly selects $\Delta_1, \Delta_2 \in \mathbb{Z}_q$, $\Delta_1 = \gamma(w_{2,i} - w_{1,i})$ and $\Delta_2 = \theta(w_{4,i} - w_{3,i})$ and sets $n = N_0 + l + 3$. The public key PK and master secret key MSK can be calculated as follows:

$PK = (H, \{W_{1,i}, W_{2,i}, W_{3,i}, W_{4,i}, Z_{1,i}, Z_{2,i}, Z_{3,i}, Z_{4,i}\}_{i=1}^{n}, g, g_1 = g^{\Delta_1}, g_2 = g^{\Delta_2}, E = e(g, g_3), V, Y)$

$MSK = (g_3, \{w_{1,i}, w_{2,i}, w_{3,i}, w_{4,i}, z_{1,i}, z_{2,i}, z_{3,i}, z_{4,i}\}_{i=1}^{n}, \gamma, \theta)$

where $1 \leq i \leq n$, $W_{1,i} = g^{w_{1,i}}$, $W_{2,i} = g^{w_{2,i}}$, $W_{3,i} = g^{w_{3,i}}$, $W_{4,i} = g^{w_{4,i}}$, $Z_{1,i} = g^{z_{1,i}}$, $Z_{2,i} = g^{z_{2,i}}$, $Z_{3,i} = g^{z_{3,i}}$, $Z_{4,i} = g^{z_{4,i}}$, and $V = g^{\gamma}$, $Y = g^{\theta}$.

(2) *IndGen*$(PK, W, \mathbb{A})$: Let $W = \{w_1, \cdots, w_{l_1}\}$, where $l_1 \leq l$, represent the keywords extracted from the EHR file, while $f = \sum_{t=0}^{l_1} \eta_t x^t$ represents a polynomial such that $H(w_1), \cdots, H(w_{l_1})$ are the roots of f. Assume that the access policy $\mathbb{A}$ includes $n_0 \leq N_0$ wildcards, which occur at positions $J = \{\omega_1, \cdots, \omega_{n_0}\}$. Meanwhile, $n_i \leq N_i$ attributes with i^{th} values appear at positions $S_i = \{s_{i0}, s_{i1}, \cdots, s_{in_i}\}$, $i \in [1, m]$. Based on the Viètes formulas $\prod_{\omega_j \in J}(i - \omega_j) = \sum_{h=0}^{n_0} a_h i^h$, we perform the computations for the coefficients $a_0, a_1, \cdots, a_{n_0}$ as follows:

$$\begin{aligned} a_0 &= -(\omega_1 \omega_2 \cdots \omega_{n_0}) \\ &\cdots \\ a_{n_0 - 1} &= -(\omega_1 + \omega_2 + \cdots + \omega_{n_0}) \\ a_{n_0} &= 1 \end{aligned}$$

Next, we perform the following calculations for the computers:

$$\Gamma_i = \sum_{x \in S_i} \prod_{\omega_j \in J} (x - \omega_j), i \in [1, m]$$

$$\Gamma = \sum_{i=1}^{m} \Gamma_i$$

The index vector is

$$\vec{x} = (a_0, a_1, \cdots, a_{n_0}, 0_{n_0+1}, \cdots, 0_{N_0}, \Gamma, \eta_0, \cdots, \eta_{l_1}, 0_{l_1+1}, \cdots, 0_l).$$

The algorithm randomly picks numbers $s_1, s_2, \alpha, \beta \in \mathbb{Z}_q$ and incorporates them into the generation of the index ciphertext as follows:

$$\begin{aligned} &C = E^{s_2}, C_0 = g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1} \\ &\{C_{1,i}, C_{2,i}\}_{i=1}^{n} = \{W_{1,i}^{s_1} Z_{1,i}^{s_2} V^{x_i \alpha}, W_{2,i}^{s_1} Z_{2,i}^{s_2} V^{x_i \alpha}\}_{i=1}^{n} \\ &\{C_{3,i}, C_{4,i}\}_{i=1}^{n} = \{W_{3,i}^{s_1} Z_{3,i}^{s_2} Y^{x_i \beta}, W_{4,i}^{s_1} Z_{4,i}^{s_2} Y^{x_i \beta}\}_{i=1}^{n}, \end{aligned}$$

The complete components of a index ciphertext are shown below:

$$CT = (C, C_0, C_1, C_2, \{C_{1,i}, C_{2,i}, C_{3,i}, C_{4,i}\}_{i=1}^{n})$$

The data owner uploads the encrypted EHR file f_i, where $i \in [1, D]$, and sends the encrypted index ciphertext CT to the cloud service, which returns the storage address. Subsequently, the data owner embeds CT and its storage address $Address.CT$ into a transaction T_x and broadcasts the signed transaction T_x to the entire blockchain system in the form of a transaction T_x. Furthermore, miners store verified transaction records on the blockchain. As shown in Table 2, the data structure of the blockchain consists of a block header and a transaction sheet. A block header comprises four parts: block id ID, block size $size$, preblock hash $hash$, and timestamp t. A transaction sheet consists of three parts: producer id ID_{DO}, previous block sig δ_{DO}, and transaction T_x.

TABLE 2. Blockchain data structure.

Block Header				Transaction Sheet		
Block Id	Block Size	Preblock Hash	Times-tamp	Producer Id	Previous Block Sig	Trans-action
ID	$size$	$hash$	t	ID_{DO}	δ_{DO}	T_x, T_y

(3) *TrapGen*(PK, MSK, S', Q): Firstly, we assume that a medical institution is added to the system with its own attributes set S'. S' contains $n'_i \leq N_i$ attributes with i^{th} value, which appear at positions $S'_i = \{s'_{i1}, s'_{i2}, \cdots, s'_{in'_i}\}$, $i \in [1, m]$. By means of the Viète's formulas, we can perform the calculations listed below:
If $S'_i \neq \emptyset$, $x_{ip} = \sum_{k \in s'_i} k^p$, $p \in [0, N_0]$, $i \in [1, m]$.
Otherwise, $x_{ip} = 0$, $p \in [0, N_0]$, $i \in [1, m]$.
This creates a lot of vectors $\overrightarrow{x_1}, \overrightarrow{x_2}, \cdots, \overrightarrow{x_m}$, as follows:

$$\begin{aligned}\overrightarrow{x_1} &= (x_{10}, x_{11}, \cdots, x_{1N_0})\\ \overrightarrow{x_2} &= (x_{20}, x_{21}, \cdots, x_{2N_0})\\ &\cdots\\ \overrightarrow{x_m} &= (x_{m0}, x_{m1}, \cdots, x_{mN_0})\end{aligned}$$

Then, we compress the vector as follows:

$$\begin{aligned}\overrightarrow{v_m} &= (\sum_{i=1}^{m} \overrightarrow{x_i})\\ &= (v_0, v_1, \cdots, v_{N_0}, -1, 0_{N_0+2}, \ldots, 0_l)\end{aligned}$$

Secondly, in order to create the trapdoor related to search keywords $Q = \{w_1, \cdots, w_s\}$, where $s \leq l$, and the date user, such as the medical institution, selects random numbers, as follows: $r_{1,i}, r_{2,i}, f_1 \in \mathbb{Z}_q$, where $1 \leq \mathrm{i} \leq \mathrm{n}$, and calculates $q_j = s^{-1} \sum_{t=1}^{s} H(w_t)^j$, where $0 \leq \mathrm{j} \leq l$.

$$\begin{aligned}q_0 &= \frac{1}{s}(H(w_1)^0 + H(w_2)^0 + \cdots + H(w_k)^0)\\ q_1 &= \frac{1}{s}(H(w_1)^1 + H(w_2)^1 + \cdots + H(w_k)^1)\\ &\cdots\\ q_j &= \frac{1}{s}(H(w_1)^j + H(w_2)^j + \cdots + H(w_k)^j)\end{aligned}$$

Meanwhile, the vectors are set as follows:

$$\overrightarrow{v_q} = (0_0, \cdots, 0_{N_0+1}, q_0, \ldots, q_l)$$

Therefore, the search vector is

$$\begin{aligned}\overrightarrow{v} &= \overrightarrow{v_m} + \overrightarrow{v_q}\\ &= (v_0, v_1, \cdots, v_{N_0}, -1, q_0, \ldots, q_l).\end{aligned}$$

Next, *MSK* is utilized for participation in trapdoor generation, as follows:

$$\begin{aligned}\{T_{1,i}, T_{2,i}\}_{i=1}^{n} &= \{g^{-\gamma r_{1,i}} g^{f_1 w_{2,i} v_i}, g^{\gamma r_{1,i}} g^{-f_1 w_{1,i} v_i}\}_{i=1}^{n}\\ \{T_{3,i}, K_{4,i}\}_{i=1}^{n} &= \{g^{-\theta r_{2,i}} g^{f_1 w_{4,i} v_i}, g^{\theta r_{2,i}} g^{-f_1 w_{3,i} v_i}\}_{i=1}^{n}\\ T_0 &= g_3 \Pi_{i=1}^{n} T_{1,i}^{-z_{1,i}} T_{2,i}^{-z_{2,i}} T_{3,i}^{-z_{3,i}} T_{4,i}^{-z_{4,i}}\\ \{T_{5,i}, T_{6,i}\}_{i=1}^{n} &= \{g^{r_{1,i}}, g^{r_{2,i}}\}_{i=1}^{n}\end{aligned}$$

The complete components of a trapdoor are listed below:

$$T = (\{T_{1,i}, T_{2,i}, T_{3,i}, T_{4,i}\}_{i=1}^{n}, T_0, \{T_{5,i}, T_{6,i}\}_{i=1}^{n})$$

Firstly, the trapdoor T is embedded into the transaction T_y. Secondly, the transaction T_y is signed and broadcasted to the entire blockchain system in the form of a transaction T_y. Finally, miners store the verified transaction $\{T_y = T\}$ on the blockchain.

(4) *Search*(CT, T): During the search phase, the search algorithm is executed by nodes on the blockchain. On one hand, the information pertaining to the data file and search keywords remains secure and is not disclosed to the blockchain or cloud server during the search process. When a user creates transaction T_y, which is associated with their own information, the nodes on the blockchain calculate transaction τ, including the successfully matched CT. After broadcasting the signed transaction, the user receives the reward d from transaction T_y. On the other hand, if transaction τ has not been recorded on the blockchain, the user can create a new transaction to reclaim the rewards from a previous transaction T_y. In this algorithm, the equation verified by the nodes on the blockchain is as follows:

$$\begin{aligned}&C \cdot \prod_{i=1}^{n} e(C_1, T_{5,i}) e(C_2, T_{6,i})\\ &\overset{?}{=} e(C_0, T_0) \cdot \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i}, T_{j,i})\end{aligned}$$

If the above expression holds true, the blockchain will return the storage address $Address.CT$ of CT to the data user; otherwise, it will return 0.

B. CORRECTNESS VALIDATION

The algorithm's correctness is verified as listed below:
Left formula:

$$\begin{aligned}&C \cdot \prod_{i=1}^{n} e(C_1, T_{5,i}) e(C_2, T_{6,i})\\ &= e(g, g_3)^{s_2} \cdot \prod_{i=1}^{n} e(g^{\Delta_1 s_1}, g^{r_{1,i}}) e(g^{\Delta_2 s_1}, g^{r_{2,i}})\\ &= e(g, g_3)^{s_2} \cdot e(g, g)^{s_1 \gamma \sum_{i=1}^{n} r_{1,i}(w_{2,i} - w_{1,i})}\\ &\quad \cdot e(g, g)^{s_1 \theta \sum_{i=1}^{n} r_{2,i}(w_{4,i} - w_{3,i})}\end{aligned}$$

Right formula:

$$\begin{aligned}&e(C_0, T_0)\\ &\quad = e(g^{s_2}, g_3 \Pi_{i=1}^{n} T_{1,i}^{-z_{1,i}} T_{2,i}^{-z_{2,i}} T_{3,i}^{-z_{3,i}} T_{4,i}^{-z_{4,i}})\\ &\quad = e(g, g_3)^{s_2} \cdot e(g, \Pi_{i=1}^{n} T_{1,i}^{-z_{1,i}} T_{2,i}^{-z_{2,i}} T_{3,i}^{-z_{3,i}} T_{4,i}^{-z_{4,i}})^{s_2}\end{aligned}$$

$$\begin{aligned}&e(C_{1,i}, T_{1,i})\\ &\quad = e(W_{1,i}^{s_1} Z_{1,i}^{s_2} V^{x_i \alpha}, g^{-\gamma r_{1,i}} g^{f_1 w_{2,i} v_i})\\ &\quad = e(g^{w_{1,i} s_1} g^{z_{1,i} s_2} g^{\gamma x_i \alpha}, g^{-\gamma r_{1,i}} g^{f_1 w_{2,i} v_i})\\ &\quad = e(g, g)^{-s_1 \gamma w_{1,i} r_{1,i}} \cdot e(g, g)^{-r_{1,i} \gamma^2 x_i \alpha}\\ &\qquad \cdot e(g, T_{1,i})^{z_{1,i} s_2} e(g, g)^{s_1 f_1 w_{1,i} w_{2,i} v_i}\\ &\qquad \cdot e(g, g)^{s_2 \gamma \alpha f_1 w_{2,i} x_i v_i}\end{aligned}$$

TABLE 3. Equations description.

Equations	Descriptions
$f = \sum_{t=0}^{l_1} \eta_t x^t$	f represents the polynomial equation which generates the coefficient of the extracted keywords, where η_t represents the coefficients of each dimension between the $N_0 + 2$-th and $N_0 + 2 + l$-th dimensions of the generated index vector $\overrightarrow{x}$. The hashed keywords are taken as the root x of f. In this equation, $0 \leq t \leq l_1$.
$\prod_{\omega_j \in J} (i - \omega_j) = \sum_{h=0}^{n_0} a_h i^h$	$\prod_{\omega_j \in J} (i - \omega_j) = \sum_{h=0}^{n_0} a_h i^h$ represents a polynomial equation related to the set of wildcard positions J, where a_h represents the coefficients for each dimension between the 0-th and n_0th dimensions of the generated index vector $\overrightarrow{x}$. In this equation, $0 \leq h \leq n_0$.
$\Gamma_i = \sum_{x \in S_i} \prod_{\omega_j \in J} (x - \omega_j)$	Γ_i represents the calculation parameters for each element in the policy attribute value position set S_i, where ω_j denotes an element belonging to the wildcard position set J and x represents an element belonging to the policy attribute set S_i. In this equation, $1 \leq i \leq m$.
$\Gamma = \sum_{i=1}^{m} \Gamma_i$	Γ is the cumulative sum of the calculation parameters Γ_i, which helps reduce the length of the index vector $\overrightarrow{x}$. Γ is placed at the $N_0 + 1$ dimension of the index vector $\overrightarrow{x}$. In this equation, $1 \leq i \leq m$.
$x_{ip} = \sum_{k \in s_i'} k^p$	$x_{i,p}$ represents the calculation parameter for each element in the user's attribute value position set S_i', where k belongs to the elements in the user's attribute position set S_i'. The vector $\overrightarrow{v_m}$ represents the coefficients between the 0 and $N_0 + 1$ dimensions of the search vector $\overrightarrow{v}$, where $\overrightarrow{v_m}$ is multiplied by the sum of $x_{i,p}$. In this equation, $p \in [0, N_0]$, $1 \leq i \leq m$.
$q_j = s^{-1} \sum_{t=1}^{s} H(w_t)^j$	q_j represents the polynomial equation used to generate the root of the query keyword, where $H(w_t)$ represents the hashed query keyword. Meanwhile, q_j is used as the coefficient between the $N_0 + 2$-th and $N_0 + 2 + l$-th dimensions of the search vector $\overrightarrow{v}$. In this equation, $1 \leq t \leq s$, $0 \leq j \leq l$.

$$
\begin{aligned}
&e(C_{2,i}, T_{2,i})\\
&= e(W_{2,i}^{s_1} Z_{2,i}^{s_2} V^{x_i\alpha}, g^{\gamma r_{1,i}} g^{-f_1 w_{1,i} v_i})\\
&= e(g^{w_{2,i}s_1} g^{z_{2,i}s_2} g^{\gamma x_i \alpha}, g^{\gamma r_{1,i}} g^{-f_1 w_{2,i} v_i})\\
&= e(g, g)^{s_1 \gamma w_{2,i} r_{1,i}} \cdot e(g, g)^{r_{1,i}\gamma^2 x_i \alpha}\\
&\quad \cdot e(g, T_{2,i})^{z_{2,i}s_2} \cdot e(g, g)^{-s_1 f_1 w_{1,i} w_{2,i} v_i}\\
&\quad \cdot e(g, g)^{-s_2 \gamma \alpha f_1 w_{1,i} x_i v_i}
\end{aligned}
$$

$$
\begin{aligned}
&e(C_{3,i}, T_{3,i})\\
&= e(W_{3,i}^{s_1} Z_{3,i}^{s_2} Y^{x_i\beta}, g^{-\theta r_{2,i}} g^{f_1 w_{4,i} v_i})\\
&= e(g^{w_{3,i}s_1} g^{z_{3,i}s_2} g^{\theta x_i \beta}, g^{-\theta r_{2,i}} g^{f_1 w_{3,i} v_i})\\
&= e(g, g)^{-s_1 \theta w_{3,i} r_{2,i}} \cdot e(g, g)^{-r_{2,i}\theta^2 x_i \beta}\\
&\quad \cdot e(g, T_{3,i})^{z_{3,i}s_2} \cdot e(g, g)^{s_1 f_1 w_{3,i} w_{4,i} v_i}\\
&\quad \cdot e(g, g)^{s_2 \theta \beta f_1 w_{4,i} x_i v_i}
\end{aligned}
$$

$$
\begin{aligned}
&e(C_{4,i}, T_{4,i})\\
&= e(W_{4,i}^{s_1} Z_{4,i}^{s_2} Y^{x_i\beta}, g^{\theta r_{2,i}} g^{-f_1 w_{3,i} v_i})\\
&= e(g^{w_{4,i}s_1} g^{z_{4,i}s_2} g^{\theta x_i \beta}, g^{\theta r_{2,i}} g^{-f_1 w_{3,i} v_i})\\
&= e(g, g)^{s_1 \theta w_{4,i} r_{2,i}} \cdot e(g, g)^{r_{2,i}\theta^2 x_i \beta}\\
&\quad \cdot e(g, T_{2,i})^{z_{4,i}s_2} \cdot e(g, g)^{-s_1 f_1 w_{3,i} w_{4,i} v_i}\\
&\quad \cdot e(g, g)^{-s_2 \theta \beta f_1 w_{3,i} x_i v_i}
\end{aligned}
$$

Then, we have the following formula

$$
\begin{aligned}
&e(C_0, T_0) \cdot \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i}, T_{j,i})\\
&\quad = e(g, g_3)^{s_2} \cdot e(g, g)^{s_1 \gamma \sum_{i=1}^{n} r_{1,i}(w_{2,i} - w_{1,i})}\\
&\quad \cdot e(g, g)^{s_1 \theta \sum_{i=1}^{n} r_{2,i}(w_{4,i} - w_{3,i})}\\
&\quad \cdot e(g, g)^{f_1 s_2 \alpha \gamma \sum_{i=1}^{n} (w_{2,i} - w_{1,i}) x_i v_i}\\
&\quad \cdot e(g, g)^{f_1 s_2 \beta \theta \sum_{i=1}^{n} (w_{4,i} - w_{3,i}) x_i v_i}
\end{aligned}
$$

Therefore, the correctness verification is as follows:

$$
\begin{aligned}
&C \cdot \prod_{i=1}^{n} e(C_1, T_{5,i}) e(C_2, T_{6,i})\\
&= e(g, g_3)^{s_2} \cdot e(g, g)^{s_1 \gamma \sum_{i=1}^{n} r_{1,i}(w_{2,i} - w_{1,i})}\\
&\quad \cdot e(g, g)^{s_1 \theta \sum_{i=1}^{n} r_{2,i}(w_{4,i} - w_{3,i})}\\
&= e(C_0, T_0) \cdot \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i}, T_{j,i})\\
&= e(g, g_3)^{s_2} \cdot e(g, g)^{s_1 \gamma \sum_{i=1}^{n} r_{1,i}(w_{2,i} - w_{1,i})}\\
&\quad \cdot e(g, g)^{s_1 \theta \sum_{i=1}^{n} r_{2,i}(w_{4,i} - w_{3,i})}\\
&\quad \cdot e(g, g)^{f_1 s_2 \alpha \gamma \sum_{i=1}^{n} (w_{2,i} - w_{1,i}) x_i v_i}\\
&\quad \cdot e(g, g)^{f_1 s_2 \beta \theta \sum_{i=1}^{n} (w_{4,i} - w_{3,i}) x_i v_i}
\end{aligned}
$$

When the medical institution's attributes satisfy the access policy formulated by the hospital and the search keywords match the index, that is to say, the result of $< x, v >= 0$, the left formula will be equal to the right formula. Therefore, when the data user obtains the storage address $Address.CT$ of CT, they can match $Address.CT$ with $Address.f_i$ and then return $Address.f_i$ to the cloud server. The cloud service will return the encrypted EHR file to the data user upon receiving the successfully matched $Address.f_i$. Here, $i \in [1, D]$.

C. NOTATIONS AND EQUATIONS DESCRIPTION

The equations and notations used in this article are defined in Table 3 and Table 4.

V. SECURITY PROOF

As shown in Figure 3, the proposed scheme separates the challenge games into five categories to demonstrate the security of the proposed scheme. Due to the impossibility of directly simulating challenges between $Game_0$ and $Game_4$, it is necessary to employ simulation conversion from intermediate games. That is to say, the proposed scheme utilizes the mixed argumentative property of information-theoretic argumentation principles. Firstly, the five challenge games $Game_0$, $Game_1$, $Game_2$, $Game_3$ and $Game_4$ are successfully

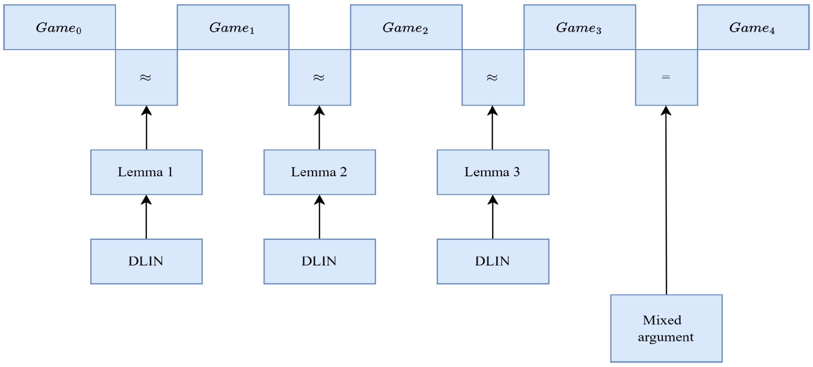


FIGURE 3. Mixed argument.

TABLE 4. Notations description.

Notations	Descriptions
U	Attributes in the universe
g	Generators of Group G
PK	The parts of the system's public key
MSK	The parts of the system's master secret key
F	File sets of the i-th data owner
W	Extracted keywords set from EHR files
Q	Set of search keywords
S'	Set of attributes for data users
$\mathbb{A}$	An access policy
CT	The index ciphertext
T	Trapdoor related to the search keywords set
$\overrightarrow{x}$	The index vector contains information about the policy and the extracted keywords
$\overrightarrow{v}$	The search vector contains information about the user's attributes and search keywords
$\overrightarrow{v_m}$	This vector contains the user's attribute information
$\overrightarrow{v_q}$	This vector contains information about the search keyword
n	The dimension of the index vector and the search vector
n_0	The actual number of wildcards
D	The number of files

constructed. Secondly, the method of proof by contradiction is employed to demonstrate that the games involved in Lemma 1, Lemma 2, Lemma 3 and Lemma 4 are indistinguishable in pairs under the DLIN assumption. Finally, it can be concluded that the advantage of the attacker being able to distinguish between $Game_0$ and $Game_4$ is negligible.

- $Game_0$. The challenge ciphertext is created according to $(\overrightarrow{x}, \overrightarrow{x})$, as listed below:

$$CT^* = (E^{s_2}, g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1}, \{W_{1,i}^{s_1}Z_{1,i}^{s_2}V^{x_i\alpha}, W_{2,i}^{s_1}Z_{2,i}^{s_2}V^{x_i\alpha}, W_{3,i}^{s_1}Z_{3,i}^{s_2}Y^{x_i\beta}, W_{4,i}^{s_1}Z_{4,i}^{s_2}Y^{x_i\beta}\}_{i=1}^{n})$$

- $Game_1$. The challenge ciphertext is created according to $(\overrightarrow{x}, \overrightarrow{0})$, as listed below:

$$CT^* = (E^{s_2}, g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1}, \{W_{1,i}^{s_1}Z_{1,i}^{s_2}V^{x_i\alpha}, W_{2,i}^{s_1}Z_{2,i}^{s_2}V^{x_i\alpha}, W_{3,i}^{s_1}Z_{3,i}^{s_2}, W_{4,i}^{s_1}Z_{4,i}^{s_2}\}_{i=1}^{n})$$

- $Game_2$. The challenge ciphertext is created according to $(\overrightarrow{x}, \overrightarrow{v})$, as listed below:

$$CT^* = (E^{s_2}, g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1}, \{W_{1,i}^{s_1}Z_{1,i}^{s_2}V^{x_i\alpha}, W_{2,i}^{s_1}Z_{2,i}^{s_2} V^{x_i\alpha}, W_{3,i}^{s_1}Z_{3,i}^{s_2}Y^{x_i\beta}, W_{4,i}^{s_1}Z_{4,i}^{s_2}Y^{x_i\beta}\}_{i=1}^{n})$$

- $Game_3$. The challenge ciphertext is created according to $(\overrightarrow{0}, \overrightarrow{v})$, as listed below:

$$CT^* = (E^{s_2}, g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1}, \{W_{1,i}^{s_1}Z_{1,i}^{s_2}, W_{2,i}^{s_1}Z_{2,i}^{s_2}, W_{3,i}^{s_1}Z_{3,i}^{s_2}Y^{x_i\beta}, W_{4,i}^{s_1}Z_{4,i}^{s_2}Y^{x_i\beta}\}_{i=1}^{n})$$

- $Game_4$. The challenge ciphertext is created according to $(\overrightarrow{v}, \overrightarrow{v})$, as listed below:

$$CT^* = (E^{s_2}, g^{s_2}, C_1 = g_1^{s_1}, C_2 = g_2^{s_1}, \{W_{1,i}^{s_1}Z_{1,i}^{s_2}V^{x_i\alpha}, W_{2,i}^{s_1}Z_{2,i}^{s_2}V^{x_i\alpha}, W_{3,i}^{s_1}Z_{3,i}^{s_2}Y^{x_i\beta}, W_{4,i}^{s_1}Z_{4,i}^{s_2}Y^{x_i\beta}\}_{i=1}^{n})$$

Lemma 1: If $|Adv_A^{Game_0} - Adv_A^{Game_1}| \leq \varepsilon$, then the DLIH assumption will be true.

Proof. we build a simulator $\mathcal{B}$ that possesses the advantage ε in solving difficulties assumption. We begin by assigning a tuple $(g, g^a, g^b, g^{ac}, g^d, Z)$ to $\mathcal{B}$. In response to the adversary $\mathcal{A}$, the following games are simulated by $\mathcal{B}$.

- *Setup*. The challenger $\mathcal{B}$ randomly selects elements $\Delta_1, \Delta_2, \gamma, \theta, \lambda \in \mathbb{Z}_q$ and $w_{1,i}, w_{2,i}, w_{3,i}, w_{4,i}, z_{1,i}, z_{2,i}, z_{3,i}, z_{4,i} \in \mathbb{Z}_q$ so that $\Delta_1 = \gamma(w_{2,i} - w_{1,i})$, $\Delta_2 = \theta(w_{4,i} - w_{3,i})$. Then, it creates

$$W_{1,i} = (g^a)^{w_{1,i}}, W_{2,i} = (g^a)^{w_{2,i}}$$
$$W_{3,i} = (g^a)^{w_{3,i}}(g^b)^{x_i\theta}, W_{4,i} = (g^a)^{w_{4,i}}(g^b)^{x_i\theta}$$
$$Z_{1,i} = g^{z_{1,i}}, Z_{2,i} = g^{z_{2,i}}$$
$$Z_{3,i} = g^{z_{3,i}}(g^b)^{x_i\theta}, Z_{4,i} = g^{z_{4,i}}(g^b)^{x_i\theta}$$
$$g_1 = g^{\Delta_1}, g_2 = g^{\Delta_2}$$
$$V = g^{\gamma}, Y = g^{\theta}$$
$$E = e(g, g)^{\lambda}$$

Finally, the challenger $\mathcal{B}$ submits the public key

$$PK = (H, g, g_1, g_2, \{W_{1,i}, W_{2,i}\}_{i=1}^{n}, \{W_{3,i}, W_{4,i}\}_{i=1}^{n}, \\ \{Z_{1,i}, Z_{2,i}\}_{i=1}^{n}, \{Z_{3,i}, Z_{4,i}\}_{i=1}^{n}, E, V, Y)$$

to the adversary $\mathcal{A}$.

- *Phase1.* The challenger $\mathcal{B}$ will calculate the trapdoor for the adversary $\mathcal{A}$.
- *Challenge.* In order to calculate the challenge ciphertext, $\mathcal{B}$ sets elements as

$$s_1 = c, s_2 = d, \alpha = \widetilde{\alpha}$$

Then, the challenger $\mathcal{B}$ provides $C_0 = g^d = g^{s_2}$, $C_1 = (g^{ac})^{\Delta_1} = g_1^{s_1}$, $C_2 = (g^{ac})^{\Delta_2} = g_2^{s_1}$ and calculates the following parameters:

$$\begin{aligned}\{C_{1,i}\}_{i=1}^{n} &= \{(g^{aw_{1,i}})^c (g^d)^{z_{1,i}} g^{x_i \gamma \widetilde{\alpha}}\}_{i=1}^{n} \\ \{C_{2,i}\}_{i=1}^{n} &= \{(g^{aw_{2,i}})^c (g^d)^{z_{2,i}} g^{x_i \gamma \widetilde{\alpha}}\}_{i=1}^{n} \\ \{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} Z^{x_i \theta}\}_{i=1}^{n} \\ \{C_{4,i}\}_{i=1}^{n} &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} Z^{x_i \theta}\}_{i=1}^{n}\end{aligned}$$

If $Z = g^{b(c+d)} g^r$, the challenger $\mathcal{B}$ simulates $Game_0$ as listed below:

$$\begin{aligned}\{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} (g^{b(c+d)} g^r)^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{3,i}^{s_1} Z_{3,i}^{s_2} Y^{x_i \beta}\}_{i=1}^{n} \\ \{C_{4,i}\}_{i=1}^{n} &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} (g^{b(c+d)} g^r)^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{4,i}^{s_1} Z_{4,i}^{s_2} Y^{x_i \beta}\}_{i=1}^{n}\end{aligned}$$

If $Z = g^{b(c+d)}$ and $r \in \mathbb{Z}_q$ is selected at random, then $\mathcal{B}$ simulates $Game_1$ with $\beta = r$ as listed below:

$$\begin{aligned}\{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} (g^{b(c+d)})^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{3,i}^{s_1} Z_{3,i}^{s_2}\}_{i=1}^{n} \\ \{C_{4,i}\}_{i=1}^{n} &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} (g^{b(c+d)})^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{4,i}^{s_1} Z_{4,i}^{s_2}\}_{i=1}^{n}\end{aligned}$$

- *Phase2.* The query of Phase 1 will be repeated by adversary $\mathcal{A}$.
- *Guess.* The adversary $\mathcal{A}$ outputs a guess bit $\theta' \in \{0, 1\}$ and wins the game if $\theta' = \theta$. If $Z = g^{b(c+d)} g^r$, the simulation algorithm is the same as $Game_0$, whereas if $Z = g^{b(c+d)}$ is a random number in G_T, the simulation algorithm is the same as $Game_1$.

Hence, the challenger $\mathcal{B}$ can solve the DLIH problem when the adversary $\mathcal{A}$ is able to distinguish between these two games.

Lemma 2: If $|Adv_A^{Game_1} - Adv_A^{Game_2}| \leq \varepsilon$, then the DLIH assumption will be true.

Proof: We build a simulator $\mathcal{B}$ with the advantage ε in solving difficulties assumptions. First, we assign a tuple $(g, g^a, g^b, g^{ac}, g^d, Z)$ to $\mathcal{B}$. In response to the adversary $\mathcal{A}$, the following games are simulated by $\mathcal{B}$.

- *Setup.* The challenger $\mathcal{B}$ randomly selects elements $\Delta_1, \Delta_2, , \gamma, \theta, \lambda \in \mathbb{Z}_q$ and $w_{1,i}, w_{2,i}, w_{3,i}, w_{4,i}, z_{1,i}, z_{2,i}, z_{3,i}, z_{4,i} \in \mathbb{Z}_q$ so that $\Delta_1 = \gamma(w_{2,i} - w_{1,i})$, $\Delta_2 = \theta(w_{4,i} - w_{3,i})$. Then, it creates

$$\begin{aligned} W_{1,i} &= (g^a)^{w_{1,i}}, W_{2,i} = (g^a)^{w_{2,i}} \\ W_{3,i} &= (g^a)^{w_{3,i}} (g^b)^{x_i \theta}, W_{4,i} = (g^a)^{w_{4,i}} (g^b)^{x_i \theta} \\ Z_{1,i} &= g^{z_{1,i}}, Z_{2,i} = g^{z_{2,i}} \\ Z_{3,i} &= g^{z_{3,i}} (g^b)^{x_i \theta}, Z_{4,i} = g^{z_{4,i}} (g^b)^{x_i \theta} \\ g_1 &= g^{\Delta_1}, g_2 = g^{\Delta_2} \\ V &= g^{\gamma}, Y = g^{\theta} \\ E &= e(g, g)^{\lambda}\end{aligned}$$

Finally, the challenger $\mathcal{B}$ submits the public key

$$PK = (H, g, g_1, g_2, \{W_{1,i}, W_{2,i}\}_{i=1}^{n}, \{W_{3,i}, W_{4,i}\}_{i=1}^{n}, \\ \{Z_{1,i}, Z_{2,i}\}_{i=1}^{n}, \{Z_{3,i}, Z_{4,i}\}_{i=1}^{n}, E, V, Y)$$

to the adversary $\mathcal{A}$.

- *Phase 1.* The challenger $\mathcal{B}$ will calculate the trapdoor for the adversary $\mathcal{A}$.
- *Challenge.* In order to calculate the challenge ciphertext, $\mathcal{B}$ sets elements as

$$s_1 = c, s_2 = d, \alpha = \widetilde{\alpha}$$

Then, the challenger $\mathcal{B}$ provides $C_0 = g^d = g^{s_2}$, $C_1 = (g^{ac})^{\Delta_1} = g_1^{s_1}$, $C_2 = (g^{ac})^{\Delta_2} = g_2^{s_1}$ and calculates the following parameters as follows:

$$\begin{aligned}\{C_{1,i}\}_{i=1}^{n} &= \{(g^{aw_{1,i}})^c (g^d)^{z_{1,i}} g^{x_i \gamma \widetilde{\alpha}}\}_{i=1}^{n} \\ \{C_{2,i}\}_{i=1}^{n} &= \{(g^{aw_{2,i}})^c (g^d)^{z_{2,i}} g^{x_i \gamma \widetilde{\alpha}}\}_{i=1}^{n} \\ \{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} Z^{x_i \theta}\}_{i=1}^{n} \\ \{C_{4,i}\}_{i=1}^{n} &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} Z^{x_i \theta}\}_{i=1}^{n}\end{aligned}$$

If $Z = g^{b(c+d)}$, then the challenger $\mathcal{B}$ simulates $Game_1$ as listed below:

$$\begin{aligned}\{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} (g^{b(c+d)})^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{3,i}^{s_1} Z_{3,i}^{s_2}\}_{i=1}^{n} \\ \{C_{4,i}\}_{i=1}^{n} &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} (g^{b(c+d)})^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{4,i}^{s_1} Z_{4,i}^{s_2}\}_{i=1}^{n}\end{aligned}$$

If $Z = g^{b(c+d)} g^r$ and $r \in \mathbb{Z}_q$ is selected at random, then $\mathcal{B}$ simulates $Game_2$ with $\beta = r$, as listed below:

$$\begin{aligned}\{C_{3,i}\}_{i=1}^{n} &= \{(g^{aw_{3,i}})^c (g^d)^{z_{3,i}} (g^{b(c+d)} g^r)^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{3,i}^{s_1} Z_{3,i}^{s_2} Y^{x_i \beta}\}_{i=1}^{n} \{C_{4,i}\}_{i=1}^{n} \\ &= \{(g^{aw_{4,i}})^c (g^d)^{z_{4,i}} (g^{b(c+d)} g^r)^{x_i \theta}\}_{i=1}^{n} \\ &= \{W_{4,i}^{s_1} Z_{4,i}^{s_2} Y^{x_i \beta}\}_{i=1}^{n}\end{aligned}$$

- *Phase2.* The query of Phase 1 will be repeated by adversary $\mathcal{A}$.
- *Guess.* The adversary $\mathcal{A}$ outputs a guess bit $\theta' \in \{0, 1\}$ and wins the game if $\theta' = \theta$. If $Z = g^{b(c+d)} g^r$, the simulation algorithm is the same as $Game_1$, whereas

TABLE 5. Functionality comparison.

Scheme	Access Policy	Wildcards	Hidden Policy	Multi-keyword	Blockchain
[14]	AND-gates on multi-valued attributes	×	Partially hidden	×	×
[17]	AND-gates on multi-valued attributes	×	Partially hidden	✓	×
[23]	AND-gates on +/-	✓	Fully hidden	✓	×
[33]	AND-gates on multi-valued attributes	✓	Partially hidden	✓	×
Ours	AND-gates on multi-valued attributes	✓	Fully hidden	✓	✓

TABLE 6. Complexity comparison.

Scheme	[14]	[17]	[23]	[33]	Ours
IndGen	$\mathcal{O}(u)E$+ $\mathcal{O}(1)E_T$	$\mathcal{O}(u)E$ +$\mathcal{O}(l_1)E_T$	$\mathcal{O}(w+l_1)E$	$\mathcal{O}(u+l_1)E$+ $\mathcal{O}(1)E_T$	$\mathcal{O}(w+l_1)E$+$\mathcal{O}(1)E_T$
Trapdoor	$\mathcal{O}(s)E$	$\mathcal{O}(u)E$+$\mathcal{O}(s)H$	$\mathcal{O}(w+s)E$	$\mathcal{O}(u+s)E$	$\mathcal{O}(w+s)E$
Search	$\mathcal{O}(1)E_T$+$\mathcal{O}(u)P$	$\mathcal{O}(1)P$	$\mathcal{O}(w+s)P$	$\mathcal{O}(s)E_T$+$\mathcal{O}(u)P$	$\mathcal{O}(w+s)P$
Ciphertext size	$\mathcal{O}(um+u)\|G\|$+$\mathcal{O}(1)\|G_T\|$	$\mathcal{O}(u)\|G\|$+$\mathcal{O}(l_1)\|G_T\|$	$\mathcal{O}(w+l_1)\|G\|$	$\mathcal{O}(um+l_1)\|G\|$+$\mathcal{O}(1)\|G_T\|$	$\mathcal{O}(w+l_1)\|G\|$ +$\mathcal{O}(1)\|G_T\|$

Notations. u: the number of attributes; m: the maximum number of attribute values ; w: the maximum number of wildcards; l_1: the number of extracted keywords; s: the number of search keywords.

if $Z = g^{b(c+d)}$ is a random number in G_T then the simulation algorithm is the same as $Game_2$.
Hence, the challenger $\mathcal{B}$ can solve the DLIH problem when the adversary $\mathcal{A}$ can successfully distinguish between these two games.
Lemma 3: If $|Adv_A^{Game_2} - Adv_A^{Game_3}| \leq \varepsilon$*, then the DLIH assumption will be true.*

- The challenge method of $Game_1$ and $Game_2$ can be used to prove the indistinguishability between $Game_2$ and $Game_3$.
Lemma 4: If $|Adv_A^{Game_3} - Adv_A^{Game_4}| \leq \varepsilon$*, then the DLIH assumption will be true.*
- The challenge method of $Game_0$ and $Game_1$ can be used to prove the indistinguishability between $Game_3$ and $Game_4$.

VI. PERFORMANCE EVALUATION

In this section, we focus on the contrast between functionality and efficiency, considering the features supported by the scheme and conducting a system analysis in order to determine the operational costs related to computation and ciphertext size. Finally, we implement all the above schemes and conduct a comprehensive evaluation of their performance.

A. FUNCTIONALITY COMPARISON

According to Table 5, the access policy of the scheme [33] and the scheme proposed in this paper support AND-gates on multi-valued with wildcards, allowing greater expression of access policies compared to the schemes proposed by [14], [17] and [23]. In addition, our scheme implements the fully hidden policy, in which neither the attribute name nor the attribute value are visible to adversaries. However, privacy leakage remains an ongoing concern, since only partially hidden policies are achieved in schemes [14], [17] and [33]. In terms of search mode, only a single-keyword search is provided in the scheme described by the authors of [14]. However, multi-keyword searchability is implemented in the schemes proposed in [17], [23], and [33] and in our proposed scheme, which greatly improves the search accuracy. In terms of the index storage, the keyword ciphertext index in schemes [14], [17], [23] and [33] are all stored in the cloud service. Nevertheless, the suggested scheme addresses the semi-trust issue presented by cloud servers by storing the index address on the blockchain.

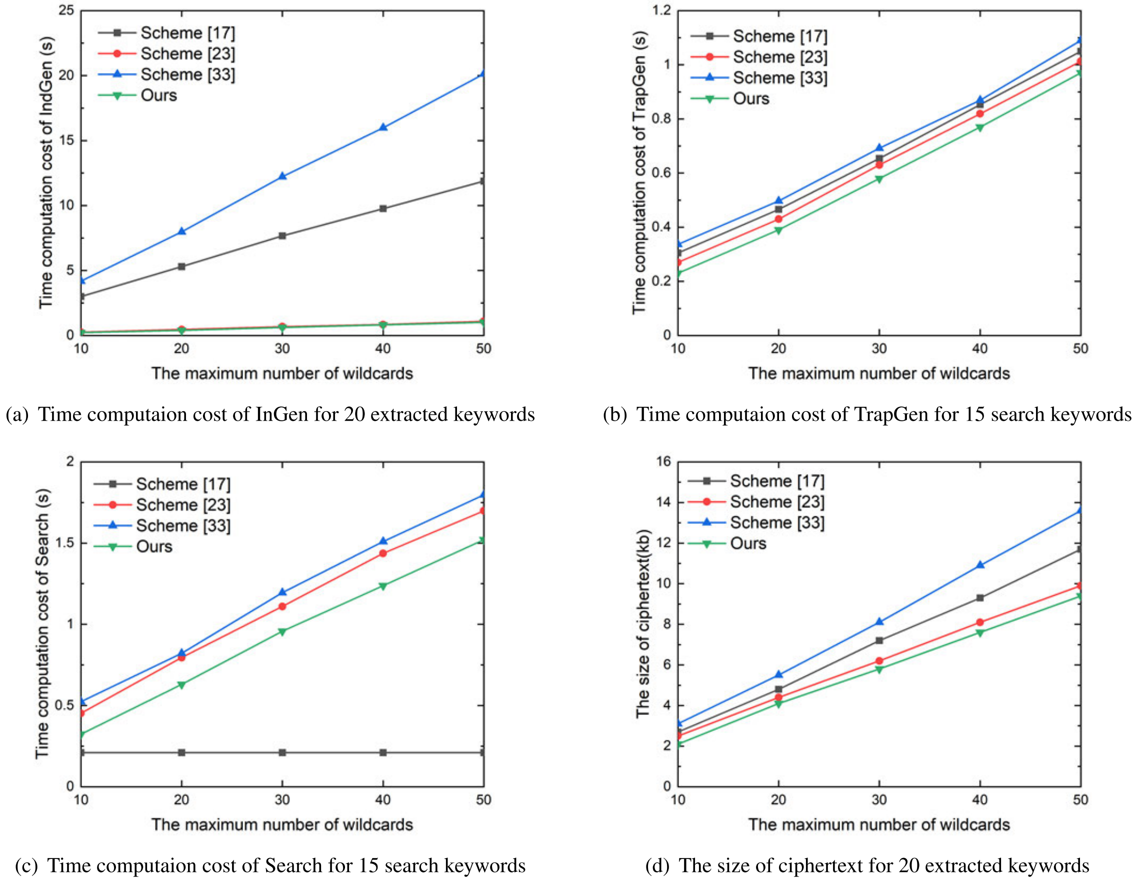
According to the above analysis, compared with other schemes, there is no doubt that our proposed scheme has all the necessary features for functionality.

B. COMPLEXITY COMPARISON

Our greatest concern is the variance between our scheme and the others described herein in terms of both ciphertext size and computational cost. As shown in Table 6, the computational cost is calculated based on the number and type of operations in each operation. To better compare computational costs, let P represent bilinear pairing on $e(G, G) \rightarrow G_T$, while H denotes the operation of hash and E and E_T denote the operation of exponentiation within groups G and G_T respectively. In this instance, $|G|$ and $|G_T|$ denote the number of bits in the elements belonging to G and G_T.

1) INDGEN

In the encryption phase, the keyword index embedded in the access policy is encrypted and then transmitted to the blockchain. The time computation complexity of the *IndGen* algorithm in the proposed scheme is $\mathcal{O}(w+l_1)E + \mathcal{O}(1)E_T$. Meanwhile, the time computation complexity of schemes [14], [17], [23] and [33] are $\mathcal{O}(u)E + \mathcal{O}(1)E_T$, $\mathcal{O}(u)E+\mathcal{O}(l_1)E_T$, $\mathcal{O}(w+l_1)E$ and $\mathcal{O}(u+l_1)|E|+\mathcal{O}(1)|E_T|$. The additional workload of the *IndGen* algorithm in the proposed scheme and [23] is solely contingent on the maximum number of wildcards and extracted keywords, rather than the number of attributes. Notably, both [14], [17] and [33] have additional operations in E_T. Hence, compared with schemes [14], [17] and [33], the proposed scheme requires less computation and compensates for the deficiencies in structure expression capability observed in schemes [23].



(a) Time computaion cost of InGen for 20 extracted keywords

(b) Time computaion cost of TrapGen for 15 search keywords

(c) Time computaion cost of Search for 15 search keywords

(d) The size of ciphertext for 20 extracted keywords

FIGURE 4. **The time and ciphertext computation cost of each algorithm.**

2) TRAPGEN

In the trapdoor generation phase, a data user, such as a medical institution, sends trapdoors associated with search keywords to the blockchain in order to conduct search operations. The time computation complexity of scheme [23] and the proposed scheme is $\mathcal{O}(w+s)E$ and the computation costs of schemes [14], [17] and [33] are $\mathcal{O}(u)E$, $\mathcal{O}(u)E + \mathcal{O}(s)H$, and $\mathcal{O}(u+s)E$. It is apparent that the proposed scheme surpasses schemes [14], [17], and [33], but aligns with scheme [23] in terms of trapdoor generation.

3) SEARCH

During the search phase, the search operation is performed by the blockchain when the attributes owned by the medical institution satisfy the access policy formulated by the hospital. The time computation complexity of scheme [23] and the proposed scheme is $\mathcal{O}(w+s)P$. The overheads of schemes [14], [17] and [33] are $\mathcal{O}(1)E_T + \mathcal{O}(u)P$, $\mathcal{O}(1)P$ and $\mathcal{O}(s)E_T + \mathcal{O}(u)P$. Hence, the proposed scheme is superior to schemes [14] and [33] and achieves a similar performance to scheme [23] but is less efficient than scheme [17].

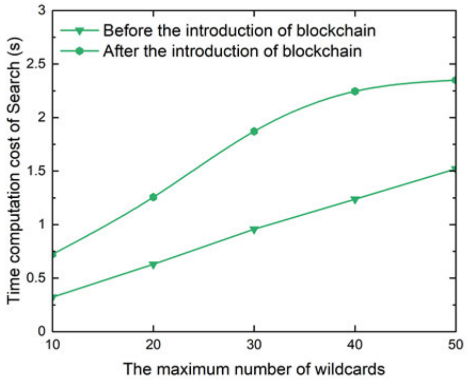
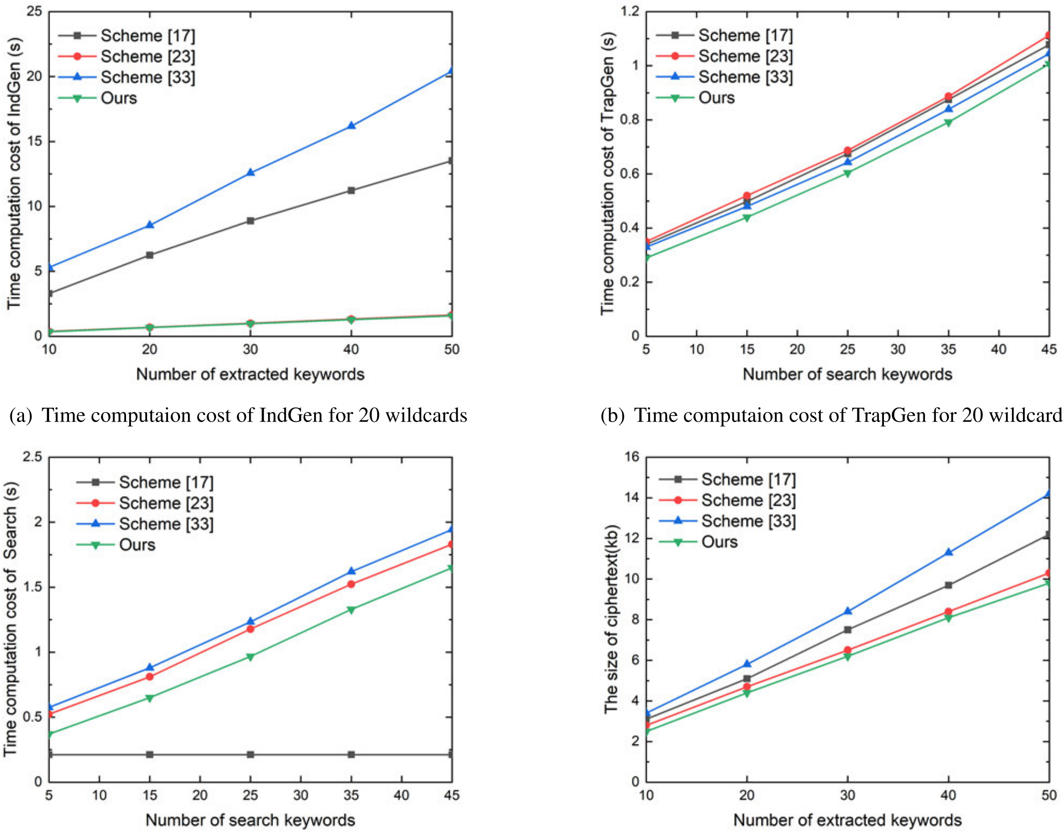


FIGURE 5. **The cost of search time computaion in the proposed scheme.**

4) CIPHERTEXT SIZE

According to Table 6, the size of ciphertext in the proposed scheme is $\mathcal{O}(w+l_1)|G| + \mathcal{O}(1)|G_T|$, while the sizes of ciphertext in the other schemes described in [14], [17], [23], and [33] are $\mathcal{O}(um+u)|G| + \mathcal{O}(1)|G_T|$, $\mathcal{O}(u)|G| + \mathcal{O}(l_1)|G_T|$, $\mathcal{O}(w+l_1)|G|$ and $\mathcal{O}(um+l_1)|G| + \mathcal{O}(1)|G_T|$, respectively. The size of ciphertext in the schemes [14] and [33] have a multiplying effect, meaning that their sizes increase with the number of attributes. Furthermore, the size of ciphertext for both schemes [23] and the proposed scheme is solely contingent on the quantity of wildcard characters and extracted keywords. As we all know, wildcards are few in number under ordinary circumstances. It is a pity



(a) Time computaion cost of IndGen for 20 wildcards

(b) Time computaion cost of TrapGen for 20 wildcards

(c) Time computaion cost of Search for 20 wildcards

(d) The size of ciphertext for 20 wildcards

FIGURE 6. The time and ciphertext computation cost of each algorithm.

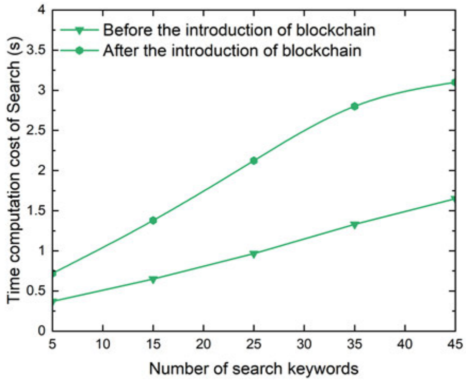


FIGURE 7. The cost of search time computaion in the proposed scheme.

that the scheme [23] only supports the access structure of AND-gates on +/− attributes with wildcards, resulting in insufficient access structure expression. In conclusion, the size of ciphertext in the proposed scheme is smaller than that of other schemes [14], [17] and [33] but similar to that of scheme [23].

C. EXPERIMENTAL COMPARISON

Our experiment was run on Windows 10 (64-bit OS, Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz, 8G RAM). We used the JPBC library and the actual World Request for Hospitals Database (RFH), which is viewed as the dataset. During the experiment, we selected 1000 files from RFH. An example of storing EHRs' details is described in Table 7 from the RFH. The data owner utilizes information such as name, age, sex, diseases, and other patient-related EHRs as extracted keywords, while corresponding name, age, sex, diseases, and other patient-related EHRs are used as search keywords.

TABLE 7. An example of EHRs.

Name	Age	Sex	Diseases
Alice	19	Male	Rhinitis
Bob	20	Male	Cataract
David	27	Female	Cardiopathy
Mary	28	Female	Pinkeye

Based on the complexity comparison, it is clear that the scheme's efficiency depends solely on the maximum number of wildcards and extracted keywords. For comparison, let us consider the worst case, in which the maximum number of wildcards matches the number of attributes. In this case, we establish that the query keyword set will be encompassed within the extracted keyword set. In light of the above factors, we select schemes [17], [23] and [33] for comparison. This section also conducts comparative experiments on the proposed scheme before and after the introduction of blockchain. The testrpc software is used to set up a local Ethereum network. The mining time is set to 0 to keep other time factors from influencing the results.

We establish the number of attributes as 20, the number of search keywords as 15, and raise the maximum number of wildcards from 10 to 50, and set the maximum number of attribute values each attribute to 10. Figure 4(a) denotes the time computation overhead of IndGen generation. With the increase in wildcards, the cost of the proposed scheme is superior to that of schemes [17] and [33], but is comparable to scheme [23]. Figure 4(b) shows the time computation overhead of trapdoor generation. As the maximum number of wildcards increases, the proposed scheme outperforms schemes [17], [23], and [33] in terms of cost. The time computation overhead of the search is represented in Figure 4(c). As the maximum number of wildcards increases, the time computation cost of the proposed scheme surpasses that of schemes [23] and [33], but is not as efficient as scheme [17]. In Figure 4(d), the storage cost of ciphertext size is depicted. As the maximum number of wildcards increases, we observe that the size of ciphertext in the proposed scheme outperforms that of schemes [17] and [33], but is comparable to [23]. Thus, with a fixed number of extracted keywords and search keywords, the proposed scheme exhibits superior advantages in terms of the index algorithm, trapdoor algorithm, and ciphertext size.

Furthermore, as shown in Figure 5, although the introduced blockchain in the proposed scheme incurs a higher computational cost of search time, the inclusion of blockchain protects the integrity of the search results and the security of the proposed scheme. More importantly, as the maximum number of wildcards increases, the computational cost growth rate of search time gradually decreased after the introduction of blockchain.

Next, we establish the maximum number of wildcards as 20, increase the number of extracted keywords from 10 to 50, and raise the number of search keywords from 5 to 45 and the maximum number of attribute values each attribute possesses to 10. The time computation cost of IndGen generation is illustrated in Figure 6(a). As the number of extracted keywords increases, the cost of the proposed scheme surpasses that of schemes [17] and [33], but remains comparable to scheme [23]. Figure 6(b) represents the time computation overhead of trapdoor generation. As the number of search keywords increases, the proposed scheme outperforms schemes [17], [23], and [33] in terms of cost. Figure 6(c) illustrates the time computation overhead of the search. As the number of search keywords increases, we observe that the time computation cost of the proposed scheme surpasses that of schemes [23] and [33], but is outperformed by scheme [17]. The storage cost of ciphertext size is depicted in Figure 6(d). As the number of extracted keywords increases, the size of ciphertext in the proposed scheme surpasses that of schemes [17], [23], and [33]. Thus, with a fixed maximum number of wildcards, the proposed scheme exhibits superior advantages in terms of the index algorithm, trapdoor algorithm, and ciphertext size.

Furthermore, as shown in Figure 7, although the introduction of blockchain in our scheme results in higher computational cost for search time compared to that observed prior to the introduction of blockchain, it guarantees the integrity of the search results and the security of the scheme. In addition, as the number of search keywords increases, the computational cost growth rate of search time gradually decreased after the introduction of blockchain.

On the whole, the proposed scheme offers significant advantages in terms of the Index algorithm, Trapdoor algorithm, Ciphertext size, and the integrity of search results.

VII. CONCLUSION

The EHR system can effectively improve the utilization rate of EHRs among different hospitals. In this paper, we have presented a blockchain-aied attribute-based searchable scheme with the properties of inner product predicate. Our scheme not only supports fully hidden fine-grained access policies with wildcards but also provides a multi-keyword search function on ciphertext data. Crucially, the size of ciphertext in the proposed scheme is determined by the maximum number of wildcards and extracted keywords, rather than the number of attributes. In other words, the size of ciphertext is more advantageous, making it highly suitable for certain lightweight medical devices. Additionally, due to the existence of the blockchain, the proposed scheme effectively avoids the semi-trusted behavior of cloud services while ensuring the reliability of the search results. The proof of security is provided by DLIN assumptions. The performance evaluation indicates that the proposed scheme is better suited for lightweight devices in EHR systems.

CONFLICTS OF INTEREST

The authors declare no competing interest.

REFERENCES

[1] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 27–33.

[2] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, "Personal health records: Definitions, benefits, and strategies for overcoming barriers to adoption," *J. Amer. Med. Inform. Assoc.*, vol. 13, no. 2, pp. 121–126, Mar. 2006.

[3] B. Tellenbach, "Identity-based cryptography," in *Trends in Data Protection and Encryption Technologies*. Cham, Switzerland: Springer, 2023, pp. 59–64.

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*. Piscataway, PA, USA: IEEE, 2005, pp. 457–473.

[5] D. Xiaoding Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy. (S&P0*, May 2000, pp. 44–55.

[6] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[7] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Int. Conf. Pairing Cryptogr.* Berlin, Germany: Springer, 2007, pp. 2–22.

[8] X. Liu, G. Yang, W. Susilo, J. Tonien, X. Liu, and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, pp. 561–574, Mar. 2021.

[9] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 522–530.

[10] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attribute-based keyword search over hierarchical data in cloud computing," *IEEE Trans. Services Comput.*, vol. 13, no. 6, pp. 985–998, Nov. 2020.

[11] J. Li, M. Wang, Y. Lu, Y. Zhang, and H. Wang, "ABKS-SKGA: Attribute-based keyword search secure against keyword guessing attack," *Comput. Standards Interfaces*, vol. 74, Feb. 2021, Art. no. 103471.

[12] Q. Huang, Q. Wei, G. Yan, L. Zou, and Y. Yang, "Fast and privacy-preserving attribute-based keyword search in cloud document services," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3348–3360, Apr. 2023.

[13] M. Padhya and D. Jinwala, "A novel approach for searchable CP-ABE with hidden ciphertext-policy," in *Information Systems Security*. Berlin, Germany: Springer, 2014, pp. 167–184.

[14] S. Qiu, J. Liu, Y. Shi, and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *Sci. China Inf. Sci.*, vol. 60, no. 5, pp. 1–12, May 2017.

[15] Y. Miao, X. Liu, K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1080–1094, May 2021.

[16] J. Gao and F. Zhou, "An encrypted cloud email searching and filtering scheme based on hidden policy ciphertext-policy attribute-based encryption with keyword search," *IEEE Access*, vol. 10, pp. 8184–8193, 2022.

[17] Y. Zhou, J. Nan, and L. Wang, "Fine-grained attribute-based multikeyword search for shared multiowner in Internet of Things," *Secur. Commun. Netw.*, vol. 2021, pp. 1–14, May 2021.

[18] A. Wu, D. Zheng, Y. Zhang, and M. Yang, "Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing," *Sensors*, vol. 18, no. 7, p. 2158, Jul. 2018.

[19] L. Yan, L. Ge, Z. Wang, G. Zhang, J. Xu, and Z. Hu, "Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment," *J. Cloud Comput.*, vol. 12, no. 1, p. 61, Apr. 2023.

[20] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. 27th Int. Conf. Theory Appl. Cryptograph. Techn.*, Istanbul, Turkey. Berlin, Germany: Springer, Apr. 2008, pp. 146–162.

[21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. 29th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: French Riviera, May/Jun. 2010, pp. 62–91.

[22] F. Meng, L. Cheng, and M. Wang, "Ciphertext-policy attribute-based encryption with hidden sensitive policy from keyword search techniques in smart city," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–22, Dec. 2021.

[23] A. Najafi, M. Bayat, and H. H. S. Javadi, "Privacy preserving attribute-based encryption with conjunctive keyword search for e-health records in cloud," *ISeCure*, vol. 13, no. 2, pp. 87–100, 2021.

[24] S. Sedghi, P. Van Liesdonk, S. Nikova, P. Hartel, and W. Jonker, "Searching keywords with wildcards on encrypted data," in *Proc. 7th Int. Conf. Secur. Cryptogr. Netw.*, Amalfi, Italy. Berlin, Germany: Springer, Sep. 2010, pp. 138–153.

[25] L. Guo, Z. Li, W.-C. Yau, and S.-Y. Tan, "A decryptable attribute-based keyword search scheme on eHealth cloud in Internet of Things platforms," *IEEE Access*, vol. 8, pp. 26107–26118, 2020.

[26] X. Zhang, D. Mu, and J. Zhao, "Attribute-based keyword search encryption for power data protection," *High-Confidence Comput.*, vol. 3, no. 2, Jun. 2023, Art. no. 100115.

[27] Z. Liu, Y. Liu, and Y. Fan, "Searchable attribute-based signcryption scheme for electronic personal health record," *IEEE Access*, vol. 6, pp. 76381–76394, 2018.

[28] C. Yin, H. Wang, L. Zhou, and L. Fang, "Ciphertext-policy attribute-based encryption with multi-keyword search over medical cloud data," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 277–284.

[29] Q. Wu, X. Ma, L. Zhang, and Y. Chen, "Expressive ciphertext policy attribute-based searchable encryption for medical records in cloud," *Int. J. Netw. Secur.*, vol. 23, no. 3, pp. 461–472, 2021.

[30] Y. Miao, J. Ma, X. Liu, F. Wei, Z. Liu, and X. A. Wang, "M2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting," *J. Med. Syst.*, vol. 40, no. 11, pp. 1–12, Nov. 2016.

[31] K. He, J. Guo, J. Weng, J. Weng, J. K. Liu, and X. Yi, "Attribute-based hybrid Boolean keyword search over outsourced encrypted data," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1207–1217, Nov. 2020.

[32] Y. Chen, W. Li, F. Gao, Q. Wen, H. Zhang, and H. Wang, "Practical attribute-based multi-keyword ranked search scheme in cloud computing," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 724–735, Mar. 2022.

[33] L. Sun and C. Xu, "Hidden policy ciphertext-policy attribute based encryption with conjunctive keyword search," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 1439–1443.

[34] S. Chen, J. Li, Y. Zhang, and J. Han, "Efficient revocable attribute-based encryption with verifiable data integrity," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10441–10451, Mar. 2024.

[35] H. Gao, Z. Ma, S. Luo, Y. Xu, and Z. Wu, "BSSPD: A blockchain-based security sharing scheme for personal data with fine-grained access control," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–20, Feb. 2021.

[36] G. Wu, S. Wang, and Z. Ning, "Blockchain-enabled privacy-preserving access control for data publishing and sharing in the Internet of Medical Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8091–8104, Jun. 2022.

[37] L. Chen, W.-K. Lee, C.-C. Chang, K.-K.-R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Gener. Comput. Syst.*, vol. 95, pp. 420–429, Jun. 2019.

[38] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, Jan. 2022.

[39] B. B. Gupta, Mamta, K.-C. Li, V. C. M. Leung, K. E. Psannis, and S. Yamaguchi, "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1877–1890, Dec. 2021.

[40] S. Niu, M. Song, L. Fang, F. Yu, S. Han, and C. Wang, "Keyword search over encrypted cloud data based on blockchain in smart medical applications," *Comput. Commun.*, vol. 192, pp. 33–47, Aug. 2022.

[41] S. Niu, L. Chen, J. Wang, and F. Yu, "Electronic health record sharing scheme with searchable attribute-based encryption on blockchain," *IEEE Access*, vol. 8, pp. 7195–7204, 2020.

[42] J. Su, L. Zhang, and Y. Mu, "BA-RMKABSE: Blockchain-aided ranked multi-keyword attribute-based searchable encryption with hiding policy for smart health system," *Future Gener. Comput. Syst.*, vol. 132, pp. 299–309, Jul. 2022.

[43] G. Han, F. Zhang, and Y. Zhang, "Blockchain-based attribute-based keyword searchable encryption for health cloud system," *Int. J. Embedded Syst.*, vol. 15, no. 6, pp. 493–504, 2022.

[44] G. Wu, B. Zhu, and J. Li, "BMKS: A blockchain based multi-keyword search scheme for medical data sharing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2022, pp. 1–7.

[45] X. Xiang and X. Zhao, "Blockchain-assisted searchable attribute-based encryption for e-health systems," *J. Syst. Archit.*, vol. 124, Mar. 2022, Art. no. 102417.

[46] S. Nakamoto. (Feb. 4, 2008). *A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

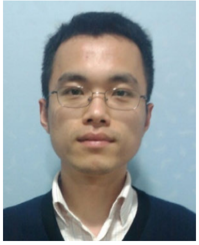
[47] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Univ. Southampton, Southampton, U.K., Ethereum Project Yellow Paper, pp. 1–32, Apr. 2014, vol. 151, no. 2014. [Online]. Available: http://ethereum.github.io/yellowpaper/paper.pdf

ZIGANG WU is currently pursuing the master's degree with the School of Information and Electronic Engineering, Zhejiang University of Science and Technology. His research interests include blockchain and cryptography.

HAIJIANG WANG received the M.S. degree from Zhengzhou University in 2013 and the Ph.D. degree from Shanghai Jiao Tong University in 2018. He currently works as a Teacher with the School of Information and Electronic Engineering, Zhejiang University of Science and Technology. His research interests include cryptography and information security, with particular emphasis on public-key encryption, attribute-based encryption, and searchable encryption.

JIAN WAN received the Ph.D. degree in computer application technology from Zhejiang University, Zhejiang, China, in 1996. He currently works as a Professor with the School of Information and Electronic Engineering, Zhejiang University of Science and Technology. His research interests include bioinformatics, service computing, and cloud computing.

LEI ZHANG received the Ph.D. degree in microbiology from the College of Life Sciences, Zhejiang University, China. She is currently a Professor with the Zhejiang University of Science and Technology, China.



JIE HUANG received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2008. He is currently an Associate Professor with the School of Information and Electronic Engineering, Zhejiang University of Science and Technology. He is the author of two books, more than 20 articles, and more than ten patents. His research interests include cloud computing, blockchain, and bioinformatics.