





## Research Article

# CD-ABSE: Attribute-Based Searchable Encryption Scheme Supporting Cross-Domain Sharing on Blockchain

Kaiyang Guo <sup>1,2</sup>, Yiliang Han <sup>1,2</sup>, Riming Wu <sup>1,2</sup> and Kai Liu <sup>1,2</sup>

<sup>1</sup>Engineering University of the PAP, Xi'an 710086, China

<sup>2</sup>Key Laboratory for Network and Information Security of the PAP, Xi'an 710086, China

Correspondence should be addressed to Yiliang Han; [yilianghan@hotmail.com](mailto:yilianghan@hotmail.com)

Received 4 May 2022; Revised 7 September 2022; Accepted 15 September 2022; Published 6 October 2022

Academic Editor: Haitao Xu

Copyright © 2022 Kaiyang Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The network security situation is grim, and the problem of “information isolated island” is becoming increasingly prominent. In view of the low efficiency and insufficient security of data cross-domain sharing in the open network environment, a searchable data sharing scheme supporting cross-domain is proposed based on attribute encryption technology. Firstly, different types of nodes on the blockchain are used to realize the data sharing of users in different domains. Secondly, the flexible ciphertext-search function is realized through the search form of keyword strategy. Moreover, the scheme adopts the mode of storage under the chain, which reduces the operation pressure of the blockchain. At the same time, according to the characteristics of the blockchain, the traceability and tamper-proof of the access process can be realized. Finally, the analysis shows that the scheme can resist quantum attack and collusive attack while avoiding complex bilinear operation and meet the security of trapdoor search and indistinguishability under chosen-plaintext attack. Compared with other searchable attribute-based encryption schemes, the scheme has certain advantages in function and performance.

## 1. Introduction

With the increasing data resources in cyberspace, the security and efficiency problems have attracted much attention. How to use information safely and efficiently to create greater value has become one of the urgent issues to be solved in this era. With the increase of the amount of individual data, there are more and more network attacks, and the data security situation is severe, which makes the cross-domain access that is already difficult to maintain permissions and low access efficiency more difficult. The failure to share data safely and efficiently will greatly reduce the value of data, resulting in a waste of resources and restricting development. However, traditional access control models, such as Discretionary Access Control, Mandatory Access Control model, and Role-Based Access Control model, have some limitations on the face of current needs. In order to ensure that the data in cyberspace can be shared more safely and efficiently, more and more scholars begin to study new access control models that are more in line with the actual needs.

Wang et al. proposed a cross-domain access control method for large organizations by applying ABAC model in distributed authoritative domain [1]. Yang and Wang proposed a new cross-domain access control model based on trust measurement [2] that can realize dynamic authorization and fine-grained access in a simple way. Shuang and Chen had built an efficient trusted cross-domain access control system by combining role mapping technology and blockchain [3]. Blockchain is used to record user roles, mapping rules, and access policies and rely on efficient smart contracts to make access decisions; Bai et al. proposed a multidomain access control service for intelligent city service system [4], which transmits data based on attribute encryption and improves the mapping efficiency through the combination of digital attribute table and B + tree. The scheme can also rely on third-party outsourcing to reduce the computational burden. Ullah et al. designed a lightweight provable cross-domain access control scheme based on the wireless body area network on the Internet of Things [5]; the computing and communication costs are reduced under the condition of ensuring security.

As a new functional public key encryption technology, it has unique advantages in data security sharing, the biggest feature of attribute-based encryption is to integrate data confidentiality and access control and determine the object of data sharing through the matching of attributes and policies. The concept of attribute-based encryption was first proposed by Sahai and Waters on the basis of identity-based encryption in 2005 [6]. Later, it is usually divided into ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE) [7]. The ciphertext-policy is formulated by the data owner, and it is more flexible in data sharing and more widely used in research and application compared with the key-policy. In 2007, Bethencourt et al. proposed the first CP-ABE scheme [8], but it does not have provable security; in the same year, Cheung et al. proposed the first scheme that can prove security under the standard model [9], but the expression ability of access structure of AND gate is limited; Waters constructed a CP-ABE scheme with flexible strategy based on linear secret sharing scheme (LSSS) in 2011 [10]. After that, many scholars put forward schemes with more perfect functions, but most of these schemes are based on bilinear map, and the complex bilinear pairing operation restricts the efficiency of the scheme. Therefore, some scholars began to try to construct attribute-based encryption schemes based on other mathematical systems. In 2012, Agrawal et al. discussed the possibility of constructing attribute-based encryption on lattice scheme [11]; in 2013, Wang proposed the CP-ABE scheme based on the learning with errors (LWE) problem on the basis of Agrawal's theory [12]. In 2015, Tan and Azmasn proposed the CP-ABE scheme based on the learning with errors over ring (RLWE) problem [13], which is significantly improved in size and efficiency compared with the scheme on the LWE problem. The research of ABE on lattice has been paid more and more attention by scholars, and the traditional problems such as access structure [14], attribute revocation, and key abuse have been deeply discussed.

With the deepening of the research on attribute-based encryption, its potential in data protection and access control has gradually attracted extensive attention in the academic community. Basu and Tripathy improved the efficiency by using CP-ABE scheme based on the security multicast requirements in the Internet of Things (IOT) [15]. In 2019, Yao and Wang protected the security of data exchanged with IOT devices based on ABE and equality testing technology [16]; Challagidat and Birje proposed a multiauthority access control scheme [17], which combined the Role Hierarchy Algorithm with the ABE, and the hierarchical access structure significantly improves the efficiency. Tian et al. applied ABE to blockchain to protect transaction privacy and realize traceability information sharing [18]. Sandoval et al. proposed a data storage method based on ABE in the cloud [19], which supports the sharing and search of encrypted data. Niu et al. used the characteristics of blockchain to improve the security of CP-ABE scheme [20]; Zhang et al. proposed an accountable data sharing model combining blockchain and ABE [21]. Based on the need of medical data, Niu et al. designed a data sharing scheme that can protect users' privacy by using ABE [22]. Kanimozhi and Victoire proposed a scheme for data sharing of the IOT based

on attribute-based encryption [23]. By performing clustering and the collected data and then encrypting it in the cloud, the confidentiality and integrity of the data are guaranteed. Li and Tan proposed an electronic certificate sharing scheme based on blockchain and attribute-based searchable encryption to achieve fine-grained access control [24].

*1.1. Security and Function Requirements.* A complete access control system should provide corresponding functions and security services to ensure data sharing among entities.

- (1) Fine-grained access control. Users can freely decide who can access the data they own and can also access the data shared by other users as needed
- (2) Data security and user privacy protection. Users' data in the process of data sharing should be safe and effective, and their personal identity information should be in a safe state
- (3) Security of index and trapdoor. During the search process, the index and trapdoor should be safe and reliable. Attackers cannot obtain more information through the index and trapdoor, nor can they destroy the system through the search process
- (4) Tailored forensics. The system shall provide certain evidence collection mechanism to ensure that the transaction has certain integrity and traceability

*1.2. Contribution.* In order to solve the problem of data sharing between different domains, this paper proposes a cross-domain access control scheme, which is based on CP-ABE to ensure data security and fine-grained access control. The cross-domain sharing of data is realized by connecting blockchains of different domains through cross-domain nodes. At the same time, the scheme also supports flexible ciphertext-search function. The main contributions of this paper are listed as follows.

- (1) Through the combination of blockchain and CP-ABE, users in the same domain and users in different domains can share data safely
- (2) The scheme supports ciphertext-search function before data access. By generating search traps in the form of keyword policy, the search of multiple keywords can be realized while ensuring privacy, which improves the flexibility of search
- (3) Using the way of ciphertext off chain storage, only a small part of the data needs to be uploaded to the blockchain, which reduces the calculation and storage pressure of the blockchain. Through encrypted storage, even if there is data leakage, it can ensure the security of information, and according to the characteristics of the blockchain, the traceability and tamper-proof of the access process can be realized
- (4) The scheme is constructed based on RLWE, without complex bilinear pairing, and has the characteristics of anti-quantum attack

1.3. *Paper Structure.* The remainder of this paper is organized as follows. In Section 2, we review some mathematical knowledge and define the security model. In Section 3, we give the system model, definition of scheme, and construction. The scheme is analyzed in Section 4, mainly including security analysis and performance analysis. Finally, we conclude our paper in Section 5.

## 2. Preliminaries

### 2.1. Lattice

*Definition 1* (lattice).  $\Lambda$  is called lattice if there are  $m$  linearly independent  $n$ -dimensional vectors in  $\Lambda$ , such that any vector in  $\Lambda$  is an integer linear combination of  $B = \{b_1, b_2, \dots, b_m\}$ , that is,  $\Lambda = \Lambda(b_1, b_2, \dots, b_m) = \{\sum_{i=1}^m s_i b_i, s_i \in \mathbf{Z}\}$ ,  $n$  is the dimension of lattice  $\Lambda$ ,  $m$  is the rank of lattice  $\Lambda$ , and  $B$  is a set of bases of lattice  $\Lambda$ .

*Definition 2* (ideal lattice). There is a ring  $R = [x]/\langle f \rangle$  and an ideal  $I \subseteq R$ ; a lattice  $\Lambda \in \mathbf{Z}^n$  is an ideal lattice if  $\Lambda$  is associated with  $I$ .

*Definition 3* (Decision R-LWE $_{d,q,\chi}$  Problem [25]). Given the security parameter  $\lambda$ , select the integer  $d, q$  based on  $\lambda$ , let  $R = \mathbf{Z}[x]/f(x)$ , where  $f(x) = x^d + 1$  and  $R_q = R/q$ . Given discrete distribution  $\chi \subset R_q$  based on  $\lambda$ , there is an unspecified challenge model  $O$  in the Decision R-LWE $_{d,q,\chi}$  Problem, that is, to determine whether the challenge model is a noisy pseudorandom sampler  $O_s$  or a real random sampler  $O_s'$  for random secret key,  $K \in R_q$ , which perform, respectively, as follows:

$O_s$ : outputs  $(\omega, \nu) = (\omega, \omega K + e) \in R_q \times R_q$ . The element  $\omega$  is uniformly random from  $R_q$ , where  $\omega \leftarrow R_q$  and the  $K \leftarrow R_q$  fixed for all samples. The element  $e \leftarrow R_q$  is a small error term that generated with a distribution  $\chi$ .

$O_s'$ : outputs truly random samples  $(\omega, \nu) \in R_q \times R_q$ .

### 2.2. Access Control Structure

*Definition 4* (Monotone Access Structure). Let  $U = \{u_1, u_2, \dots, u_n\}$  be a set of attributes. A collection  $D \subseteq U$  is monotone if  $\forall B, C : B \in D, B \subseteq C \Rightarrow C \in D$ . The sets in  $A$  are called as authorized sets, and the sets not in  $D$  are called as unauthorized sets.

*Definition 5* (linear secret sharing scheme (LSSS) [13]). The  $\Pi$  is a secret sharing scheme over a set of attributes  $U$  if the following properties are met:

- (1) All sharers have a secret sharing vector based on  $R_q$
- (2) There is a share-generating matrix  $F \in R_q^{n \times m}$  for  $\Pi$ , with row labels  $\rho(i) \in U, \forall i \in [n]$ . Given a column vector,  $\vec{v} = (s, r_2, \dots, r_m)$ , where  $s \in R_q$  is the secret to be shared and  $r_2, \dots, r_m \leftarrow R_q$  are randomly chosen.

Let  $\delta_i = F_i \times v \in R_q, i \in (1, n)$  represent attribute  $\rho(i)$ , where  $\rho(i)$  is a function from  $i$  to  $U$

Linear secret sharing scheme has linear reconstruction characteristics. Suppose that  $\Pi$  is an LSSS that represents the access structure  $A$ . Let  $A \in A$  be an authorized set, and  $I \subset \{1, \dots, n\}, I = \{i : \rho(i) \in D\}$ . There exist constants  $\{\omega_i \in R_q\}_{i \in I}$  then  $\sum_{i \in I} \delta_i \omega_i = s$  such that of  $\delta_i$  are valid shares of a secret  $s$  according to  $\Pi$ . Furthermore, these constants  $\omega_i$  can be calculated through the share-generating matrix  $F$  in polynomial time. For unauthorized sets, it cannot be calculated, that is, any information of secret sharing value cannot be obtained.

## 3. Attribute-Based Searchable Encryption Scheme Supporting Cross-Domain Sharing on Blockchain

3.1. *System Model.* The model in this scheme can be divided into three layers, such as storage layer, blockchain service layer, and application layer from bottom to top. The model is shown in Figure 1.

The storage layer is responsible for providing data storage, which is divided into blockchain data storage and IPFS (Inter Planetary File System) data storage. Blockchain data mainly includes system initialization parameters, relevant information applied by users, indexes, and initial ciphertext, etc., and these data will be stored in the form of transactions; IPFS mainly stores the encrypted data uploaded by users. In the blockchain service layer, it is mainly divided into Unit-chain and Region-chain, in which Unit-chain is mainly responsible for internal data services, including data recording and access services; The Region-chain is mainly responsible for cross domain data services between different units. Based on the weak credit environment, this model is based on the Consortium Blockchain, and only licensed nodes can operate. At the same time, the credit consensus mechanism is adopted, and the nodes with violations will be revoked and removed from the system. The nodes in this model are mainly divided into general nodes and cross domain nodes. General nodes mainly maintain blockchain services within their own units, and cross domain nodes are responsible for connecting blockchains between two different domains, providing cross domain access services, and deploying the authority on cross domain nodes to improve work efficiency and resource utilization. The application layer provides various functional applications.

The proposed system includes five entities: *Authority*, *Data Owner (DO)*, *Inter Planetary File System (IPFS)*, *Data User (DU)*, and *Blockchain*. The *Authority* is deployed on the *Blockchain*. The relationship among the entities is shown in Figure 2.

- (1) *Authority.* The authority generates the system's public parameters and master key, manages the users in the system, and constructs the private key for each user according to the user's identity and authority, then the authority generates temporary keys for

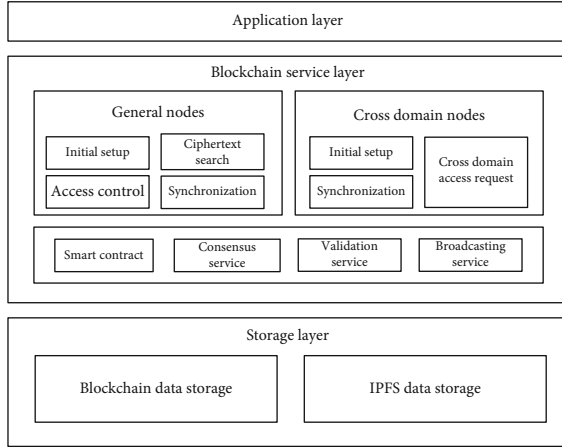


FIGURE 1: System model.

users in other domains and search traps for users during cross-domain access. We assume that the authority is completely trusted, will faithfully perform various operations, and will not disclose users' personal information. In order to facilitate operation and data processing, we deploy the authority to the cross-domain nodes of the blockchain

- (2) *Data Owner (DO)*. The data owner generates keyword index  $I_w$  based on data and encrypts data with symmetric key  $k$ , then uploads encrypted data to the IPFS. After that, DO sets the access policy of the data and encrypts the symmetric key and address, then DO uploads this ciphertext  $ct$  and index  $I_w$  to blockchain
- (3) *Inter Planetary File System (IPFS)*. The IPFS is responsible for storing data and returning an address. IPFS is honest but curious, always correctly implement the requirements put forward by all entities in the scheme, but attempts to decrypt the ciphertext content
- (4) *Data User (DU)*. The data user can access data according to their needs. Apply to the authority for a search trapdoor as needed and send it to the blockchain node. After obtaining the returned initial ciphertext, DU decrypts the ciphertext according to private key. After obtaining the address, download the corresponding data from IPFS, then DU can decrypt the data according to the symmetric key
- (5) *Blockchain*. The blockchain performs smart contract and runs algorithm, and important events in the access process will form blocks in the form of transactions and be saved in the blockchain. Due to the guarantee of trust proof of work, the node will faithfully perform operations. We assume that the entity is not completely trusted and may try to decipher user's data

**3.2. Overview of the Scheme.** Based on ABE and blockchain, the scheme realizes the data access control of users in the same domain or between different domains and can also provide ciphertext-search function. In order to ensure the trace-

ability and tamper-proof of the search process, important events in the access process will be formed into blocks in the form of transactions and stored in the blockchain. The information contained in the release record can be determined according to the specific situation. If the privacy is strong, it can be released in the form of pseudo-ID or other forms, which is not the focus of the scheme and will not be discussed too much. The access process of the scheme is shown in Figure 3. The specific contents of the scheme are as follows.

- (1) Initialize accounts, deploy smart contracts, and initialize systems
- (2) The user submits the registration information to authority, which verifies and generates the corresponding private key
- (3) DO extracts keywords from the data to be shared and generates an index  $I_w$ , then encrypts the data through a symmetric algorithm, and uploads the encrypted data to IPFS, then gets address  $L$ , then encrypts the address and symmetric key  $k$  according to own strategy to get the initial ciphertext  $ct$ , and finally, embeds  $ct$  and  $I_w$  into a transaction, and publishes it to the blockchain
- (4) When DU needs to access the data in their own domain, DU sends an application to their authority, which contains the visitor's information, keyword combination, and signature. The authority first verifies the user's identity. If the user is forged or illegal, it will refuse access; if the identity is valid, the trapdoor  $T_w'$  is generated through keyword combination and is sent to the node for search
- (5) When DU needs to access the data of other domains, they first apply to their authority. After receiving the application, the cross-domain node, as the user's agent, submits a temporary access application to the authority of the target domain. After verification, the authority of the target domain assigns a temporary private key. The private key has time or times limit when used, and then follow step (4)
- (6) According to the incentive mechanism, after receiving the search trapdoor  $T_w'$ , the node runs the algorithm for matching search that to get reward. When the keywords match, the node will return the corresponding initial ciphertext  $ct$ ; otherwise, it will return  $\perp$
- (7) When DUs receive the initial ciphertext  $ct$ , they decrypt it with their own private key. If their own attributes meet the policies formulated by the DO, it can be decrypted smoothly to obtain the address  $L$  and symmetric key  $k$ . Once decrypted successfully, the user's "wallet" will publish the access record to the blockchain; otherwise, it will return  $\perp$
- (8) Finally, DU submits the address to IPFS, downloads the corresponding encrypted data, and then, decrypts it with symmetric key  $k$  to obtain the data

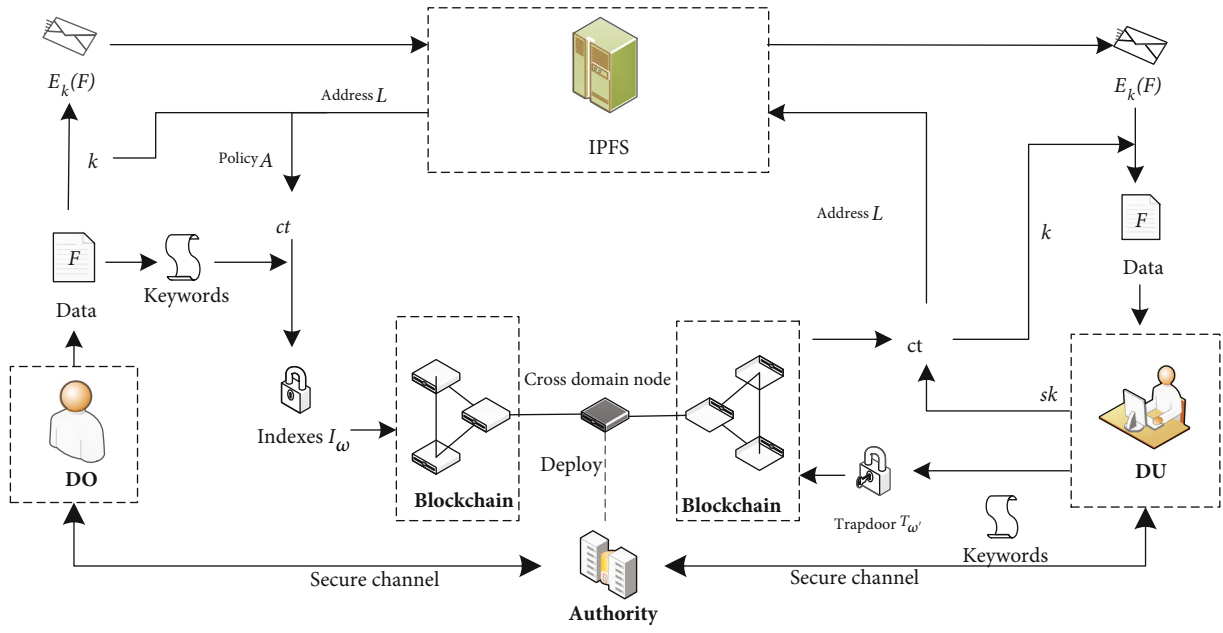


FIGURE 2: System architecture.

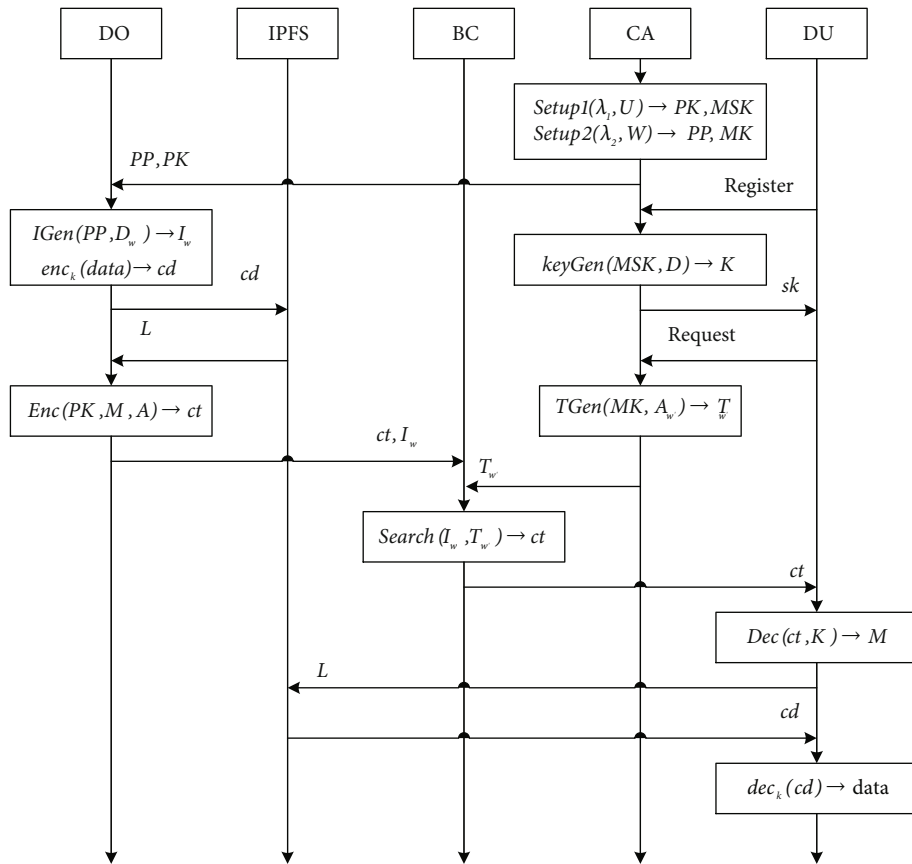


FIGURE 3: Access flow chart.

The scheme consists of the following eight algorithms.

*Setup1* ( $\lambda_1, U$ )  $\longrightarrow PK, MSK$ . The algorithm is executed by authority. Given the security parameter  $\lambda_1$ , and the collection of all attributes  $U$  in the system, this algorithm outputs public parameters  $PK$  and master secret key  $MSK$ .

*Setup2* ( $\lambda_2, W$ )  $\longrightarrow PP, MK$ . The algorithm is executed by authority. Given the security parameter  $\lambda_2$ , and the collection of all attributes  $W$  in the system, this algorithm outputs public parameters  $PP$  and master key  $MK$ .

*IGen*( $PP, D_w$ )  $\longrightarrow I_w$ . The algorithm is executed by DO. Input public parameters  $PP$ , a set of keywords  $D_w$  used to describe data. This algorithm outputs index  $I_w$ .

*Enc*( $PK, M, A$ )  $\longrightarrow ct$ . The algorithm is executed by DO. Input public parameters  $PK$ , the message  $M$  about address and symmetric key  $M = k||L$ , and user's access policy  $A$ . This algorithm outputs the ciphertext  $ct$ .

*keyGen*( $MSK, D$ )  $\longrightarrow sk$ . The algorithm is executed by authority. Input master secret key  $MSK$  and user's attribute set  $D$ . This algorithm outputs the secret key  $sk$  for the user.

*TGen*( $MK, A_{w'}$ )  $\longrightarrow T_{w'}$ . The algorithm is executed by authority. Input master key  $MK$  and user's keyword policy  $A_{w'}$ . This algorithm outputs a trapdoor  $T_{w'}$ .

*Search*( $PP, I_w, T_{w'}$ )  $\longrightarrow ct$ . The algorithm is executed by a node of blockchain. Input public parameters  $PP$ , index  $I_w$ , and a trapdoor  $T_{w'}$ ; if keywords match the corresponding data, the ciphertext  $ct$  is returned; otherwise, it return  $\perp$ .

*Dec*( $PK, ct, sk$ )  $\longrightarrow M$ . The algorithm is executed by DU. Input public parameters  $PK$ , ciphertext  $ct$ , and user's secret key  $sk$ . This algorithm outputs  $M = k||L$ , then the DU can download the data through the address  $L$  and decrypt it with the symmetric key  $k$  to obtain the data.

**3.3. Security Model.** It is assumed that the authority is a fully trusted entity. IPFS and blockchain are semitrusted entities. They will faithfully perform operations, but they may try to decipher user data; IPFS and blockchain may collude with attackers. Assuming that the channel between users and authority is a secure channel, consider the following attacker and security models.

- (1) The scheme should meet the basic data security requirements and ensure the confidentiality of the data in the sharing process. The attacker 1 mainly focuses on the security problems in the system of ABE and attempts to decrypt the encrypted data
- (2) Based on the characteristics of ABE, the scheme should be able to resist collusion attack. We define attacker 2 as malicious legitimate users, who can obtain any number of keys and attempt to collude to expand their decryption ability. It is defined that if the advantage of attacker 2 can be ignored in any polynomial time, the scheme meets the security of anticollusion attack
- (3) The scheme should meet the privacy security of the index, and the attacker should not be able to distin-

guish the index corresponding to different keywords. Define that attacker 3 attempts to obtain information from the index

- (4) The scheme should meet the privacy security of the trapdoor, and the attacker should not be able to distinguish the trapdoor corresponding to different keywords. Define that attacker 4 attempts to obtain information from the trapdoor

**Definition 6** (IND-CPA security). The definition is given by describing the game between adversary  $\mathcal{A}$  and simulator  $\mathcal{B}$ . The scheme satisfies the security of chosen-plaintext attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary  $\mathcal{A}$  selects an access structure  $A^*$  and sends it to  $\mathcal{B}$ .

*Setup.* The simulator  $\mathcal{B}$  generates public parameters  $PK$  and master key  $MSK$  and sends  $PK$  to  $\mathcal{A}$ .

*Inquiry Phase 1.* The adversary  $\mathcal{A}$  asks the simulator  $\mathcal{B}$  for the private key, but  $\mathcal{A}$ 's attribute set does not meet the access structure. The simulator runs the *KeyGen* algorithm to generate the private key and send it to  $\mathcal{A}$ .

*Challenge.* The adversary  $\mathcal{A}$  chooses two messages  $M_0, M_1 \in \{0, 1\}$  and sends them to simulator  $\mathcal{B}$ , then  $\mathcal{B}$  randomly selects  $b \in \{0, 1\}$  to calculate the challenge ciphertext and sends it to  $\mathcal{A}$ .

*Inquiry Phase 2.*  $\mathcal{A}$  asks for the key as in phase 1.

*Guess.* Adversary  $\mathcal{A}$  outputs his guess  $b'$  about  $b$ . The advantage of  $\mathcal{A}$  in this game is defined as  $\text{adv}^{\mathcal{A}} = \Pr[b' = b] - (1/2)$ .

**Definition 7** (IND-CKA security). The definition is given by describing the game between adversary  $\mathcal{A}_3$  and simulator  $\mathcal{B}_3$ . The scheme satisfies the security of chosen-keyword attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary  $\mathcal{A}_3$  selects  $D_{w_0}^*$  and  $D_{w_1}^*$  as two keywords with the same length and sends them to  $\mathcal{B}_3$ .

*Setup.* The simulator  $\mathcal{B}_3$  generates public parameters  $PP$  and master key  $MK$  and sends  $PP$  to  $\mathcal{A}_3$ .

*Inquiry Phase 1.*  $\mathcal{A}_3$  sends  $D_{wi}^*$  keywords to  $\mathcal{B}_3$ , then  $\mathcal{B}_3$  runs the *IGen*( $PP, D_w$ )  $\longrightarrow I_w$  algorithm to generate  $I_{wi}$  and send it to  $\mathcal{A}_3$ . Note that the keyword set of the query cannot be the same as the keyword set of the challenge.

*Challenge.*  $\mathcal{B}_3$  randomly selects  $b \in \{0, 1\}$  to calculate the challenge index  $I_{wb}$  and send it to  $\mathcal{A}_3$ .

*Inquiry Phase 2.*  $\mathcal{A}_3$  asks for the index as in phase 1.

*Guess.* Adversary  $\mathcal{A}_3$  outputs his guess  $b'$  about  $b$ . The advantage of  $\mathcal{A}_3$  in this game is defined as  $\text{adv}^{\mathcal{A}_3} = \Pr[b' = b] - (1/2)$ .

**Definition 8** (IND-IKGA security). The definition is given by describing the game between adversary  $\mathcal{A}_4$  and simulator  $\mathcal{B}_4$ . The scheme satisfies the security of internal-keyword

guessing attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary  $\mathcal{A}_4$  selects  $A_{w_0}$  and  $A_{w_1}$  as two keyword policies with the same length and sends them to  $\mathcal{B}_4$ .

*Setup.* The simulator  $\mathcal{B}_4$  generates public parameters  $PP$  and master key  $MK$  and sends  $PP$  to  $\mathcal{A}_4$ .

*Inquiry Phase 1.*  $\mathcal{A}_4$  sends  $A_{w_i}^*$  keyword policy to  $\mathcal{B}_4$ , then  $\mathcal{B}_4$  runs the  $TGen(MK, A_{w_i}^*) \rightarrow T_{w_i}$  algorithm to generate  $T_{w_i}$  and send it to  $\mathcal{A}_4$ . Note that the keyword set of the query cannot be the same as the keyword set of the challenge.

*Challenge.*  $\mathcal{B}_4$  randomly selects  $b \in \{0, 1\}$  to calculate the challenge keyword-policy  $A_{w_b}$  and send it to  $\mathcal{A}_4$ .

*Inquiry Phase 2.*  $\mathcal{A}_4$  asks for the trapdoor as in phase 1.

*Guess.* Adversary  $\mathcal{A}_4$  outputs his guess  $b'$  about  $b$ . The advantage of  $\mathcal{A}_4$  in this game is defined as  $\text{adv}^{\mathcal{A}_4} = \Pr[b' = b] - (1/2)$ .

### 3.4. Construction of the CD-ABSE Scheme

*3.4.1. Initialization Phase.* This phase mainly includes the initialization of the authority and the blockchain system, in which the blockchain system completes the setting of the corresponding accounts and nodes, etc. The initialization of the authority mainly includes the following two algorithms.

*Setup1*( $\lambda_1, U$ )  $\rightarrow PK, MSK$ . Given the security parameter  $\lambda_1$  and the collection of all attributes  $U$  in the system, randomly select a large prime number  $q_1 = 1 \pmod{2\lambda_1}$  and a small positive integer  $p_1$ , where  $p_1 \ll q_1$  and  $\text{gcd}(p_1, q_1) = 1$ . Let  $f(x) = (x^d + 1)$ , where  $d$  is a power of 2. Let  $R_{q_1} = \mathbb{Z}_{q_1}[x]/\langle f(x) \rangle$  be the ring of integer polynomials modulo both  $f(x)$  and  $q_1$ . Let  $\chi_1 = \chi_1(\lambda)$  be an error distribution over  $R_{q_1}$ . Select a uniformly random  $SK_0 \leftarrow R_{q_1}$  and random element  $a_1 \leftarrow R_{q_1}$ , then choose a small noise term  $e_0 \leftarrow \chi_1$ . Compute  $PK_0 = aSK_0 + pe_0 \in R_{q_1}$ . Next, select a pair of uniformly random  $(SK_i, SK_i^{-1}) \leftarrow R_{q_1}$  for each attribute in  $U$ , where  $SK_i^{-1}$  is the inverse of  $SK_i$  in  $R_{q_1}$ , and select a small noise term  $e_i \leftarrow \chi_1$ , then compute  $PK_i = SK_i + pe_i \in R_{q_1}$ . Lastly, output the public parameters  $PK = \{a, PK_0, \{PK_i\}_{i=1}^n\}$  and the master secret key  $MSK = \{SK_0, \{SK_i\}_{i=1}^n, \{SK_i^{-1}\}_{i=1}^n\}$ .

*Setup2*( $\lambda_2, W$ )  $\rightarrow PP, MK$ . Given the security parameter  $\lambda_2$ , and the collection of all keywords  $W$  in the system, randomly select a large prime number  $q_2 = 1 \pmod{2\lambda_2}$  and a small positive integer  $p_2$ , where  $p_2 \ll q_2$  and  $\text{gcd}(p_2, q_2) = 1$ . Let  $f(x) = (x^d + 1)$ , where  $d$  is a power of 2. Let  $R_{q_2} = \mathbb{Z}_{q_2}[x]/\langle f(x) \rangle$  be the ring of integer polynomials modulo both  $f(x)$  and  $q_2$ . Let  $\chi_2 = \chi_2(\lambda)$  be an error distribution over  $R_{q_2}$ . Select a uniformly random  $pk_0 \leftarrow R_{q_2}$  and random element  $a_2 \leftarrow R_{q_2}$ , then choose a small noise term  $e_{-0} \leftarrow \chi_2$ . Compute  $sk_0 = a_2pk_0 + pe_{-0} \in R_{q_2}$ . Next, select a pair of uniformly random  $(pk_i, pk_i^{-1}) \leftarrow R_{q_2}$  for each key-

word in  $W$ , where  $pk_i^{-1}$  is the inverse of  $pk_i$  in  $R_{q_2}$ , and select a small noise term  $e_{-i} \leftarrow \chi_2$ , then compute  $sk_i = pk_i + pe_{-i} \in R_{q_2}$ . Lastly, output the public parameters  $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i^{-1}\}_{i=1}^n\}$  and the master secret key  $MK = \{a_2, sk_0, \{sk_i\}_{i=1}^n\}$ .

*3.4.2. Registration Phase.* This phase mainly refers to that the user submits a registration application to the authority, and the authority runs the following algorithm to generate a key for the user.

*keyGen*( $MSK, D$ )  $\rightarrow sk$ . Input master key  $MSK$ , user's attribute set  $D$ , then choose small noise term  $e'', e_i'' \leftarrow \chi_1$ , and select a pair of uniformly random  $(t, t^{-1}) \leftarrow R_{q_1}$  for each attribute in  $D$ . Compute  $K_0 = SK_0t^{-1} + pe'' \in R_{q_1}$ ,  $K_i = SK_i^{-1}t + pe_i'' \in R_{q_1}, \forall i \in D$ ; output the secret key  $sk = (K_0, K_i)$ .

*3.4.3. Data Preparation Phase.* This phase mainly refers to the operation when the data owner shares the data, including symmetrically encrypting the data, sending the encrypted data to IPFS and obtaining the address, and then, encrypting the address and the symmetric key to obtain the ciphertext. In addition, the user also needs to generate a ciphertext index for this data. The algorithm for index generation and encryption is as follows:

*IGen*( $PP, D_w$ )  $\rightarrow I_w$ . Input public parameters  $PP$  and a keyword set  $D_w$  of data. Select a pair of uniformly random  $(t', t'^{-1}) \leftarrow R_{q_2}$ , and choose small noise term  $e_{-}', e_i' \leftarrow \chi_2$  for each keyword in  $D_w$ . Compute  $I_0 = pk_0t'^{-1} + pe_{-}' \in R_{q_2}$ ,  $I_i = pk_i^{-1}t' + pe_i' \in R_{q_2}, \forall i \in D_w$ ; output an index  $I_w = (I_0, I_i)$ .

*Enc*( $PK, M, A$ )  $\rightarrow ct$ . Input public parameters  $PK$ , the message  $M \in \{0, 1\}^n$  about  $k||L$ , and set access policy  $A = (F, \rho)$ ,  $F \in R_{q_1}^{n \times m}$  with row labels  $\rho(j) \in H, \forall j \in [n], H \in A$ . Generate a vector  $v = (s_1, r_2, \dots, r_m)$ , where  $r_2, \dots, r_m \leftarrow R_{q_1}$ , and  $s_1 \in R_{q_1}$  is the secret to be shared.  $\delta_i = F_i \times v \in R_{q_1}, i \in (1, n)$ , where  $F_i$  is the vector corresponding to  $i$ th row of  $F$ . Then, choose a uniformly random element  $r_1 \leftarrow R_{q_1}$  and noise terms  $e', e_i' \leftarrow \chi_1$ ; compute  $c_0 = PK_0r_1s_1 + M + p \cdot e' \in R_{q_1}$ ,  $c_i = ar_1PK_i\delta_i + pe_i' \in R_{q_1}$ , and output  $ct = (c_0, c_i)$ .

After completing the above steps, DO embeds the ciphertext  $ct$  and index  $I_w$  into the transaction  $T_X$  and signs it to  $T_Y$ , then broadcasts  $T_X$  to the whole blockchain. After the transaction is verified, it is recorded on the blockchain by the miner. The data structure is as shown in Table 1.

*3.4.4. Access Preparation Phase.* The user sends the data keywords to be accessed to the authority, and the authority executes the following algorithm to generate a search trapdoor for the user.

*TGen*( $MK, A_{w'}$ )  $\rightarrow T_{w'}$ . Input master key  $MK$ , a keyword set  $D_w$  of data, and set keyword policy  $A_{w'} = (F_w, \rho)$ ,  $F_w \in R_{q_2}^{n \times m}$  with row labels  $\rho(j) \in H, \forall j \in [n], H \in A_{w'}$ . Generate a vector  $v = (s_2, r_2, \dots, r_m)$ , where  $r_2, \dots, r_m \leftarrow R_{q_2}$ , and  $s_2 \in R_{q_2}$  is the secret to be shared.  $\delta_i' = F_{wi} \times v \in R_{q_2}, i \in (1,$

TABLE 1: Blockchain data structure in data preparation phase.

Identification	From	To	Action	Timestamp	Signature	Transaction
ID_1	DO	BC	Publish	Timestamp1	Sig1	$T_X, T_Y$

$n$ ), where  $F_{wi}$  is the vector corresponding to  $i$ th row of  $F_w$ . Then, choose a uniformly random element  $r_1' \leftarrow R_{q_2}$ , and noise terms  $e_{-i}' \leftarrow \chi_2$ ; compute  $T_0 = sk_0 r_2 s_2 + pe_{-i}' \in R_{q_2}$ ,  $T_i = a_2 sk_i r_2 \delta_i' + pe_{-i}' \in R_{q_2}$ , and output trapdoor  $T_w' = (T_0, T_i)$ .

**3.4.5. Search Phase.** The search phase mainly involves two parts. The first is that DU embeds trapdoor  $T_w'$  into  $T_X$ , then publishes it to the smart contract address of the blockchain, and then invokes the search contract for calculation and retrieval. After the search is completed, the blockchain returns the data to DU through the user address. The two data structures are shown in Table 2.

$Search(PP, I_w, T_w') \rightarrow ct$ . Input public parameters  $PP$ , index  $I_w$ , and trapdoor  $T_w'$ . If the set of keyword meets the keyword policy  $A_w'$  and  $I \subset \{1, \dots, n\}$ ,  $I = \{i : \rho(i) \in A_w'\}$ , compute a set of constants  $\{\omega_i \in R_{q_2}\}_{i \in I}$  with a linear reconstruction algorithm of LSSS, then  $\sum_{i \in I} \delta_i' \omega_i = s_2$ , and compute  $J = T_0 - I_0 \sum_{i \in I} I_i \omega_i T_i \bmod p$ ; if  $J = 0$ , the search is successful, and  $ct$  is returned; otherwise, it return  $\perp$ . The correctness of the successful search of the scheme is explained as follows.

$$\begin{aligned}
J' &= T_0 - I_0 \sum_{i \in I} I_i \omega_i T_i = T_0 - I_0 \sum_{i \in I} I_i \omega_i (a_2 sk_i r_2 \delta_i' + pe_{-i}') \\
&= T_0 - I_0 a_2 r_2 s_2 \sum_{i \in I} sk_i I_i - I_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i) \\
&= T_0 - I_0 a_2 r_2 s_2 \sum_{i \in I} (pk_i + pe_{-i}) (pk_i^{-1} t + pe_{-i}'') \\
&\quad - I_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i) = T_0 - I_0 a_2 r_2 s_2 \sum_{i \in I} \\
&\quad \cdot (t + pe_{-i} pk_i^{-1} t + pk_i pe_{-i}'' + p^2 e_{-i} e_{-i}'') \\
&\quad - I_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i) = T_0 - I_0 a_2 r_2 s_2 t \\
&\quad - p I_0 a_2 r_2 s_2 \sum_{i \in I} (e_{-i} pk_i^{-1} t + k_i pe_{-i}'' + pe_{-i} e_{-i}'') \\
&\quad - I_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i) = sk_0 r_2 s_2 + pe_{-i}' \\
&\quad - (pk_0 t^{-1} + pe_{-i}'') a_2 r_2 s_2 t - p T_0 a_2 r_2 s_2 \sum_{i \in I} \\
&\quad \cdot (e_{-i} pk_i^{-1} t + k_i pe_{-i}'' + pe_{-i} e_{-i}'') - T_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i) \\
&= pe_{-0} r_2 s_2 + pe_{-i}' - pe_{-i}'' a_2 r_2 s_2 t - p T_0 a_2 r_2 s_2 \sum_{i \in I} \\
&\quad \cdot (e_{-i} pk_i^{-1} t + k_i pe_{-i}'' + pe_{-i} e_{-i}'') - T_0 p \sum_{i \in I} (e_{-i}' \omega_i I_i).
\end{aligned} \tag{1}$$

If the conditions are met, then  $J = J' \bmod p = 0$ . Otherwise, there will be  $\sum_{i \in I} \delta_i' \omega_i \neq s_2$  and  $J = J' \bmod p \neq 0$ ; the access will be terminated.

The above algorithm will be executed in the smart contract, and the design of smart contract is shown in Table 3.

**3.4.6. Decryption Phase.** After receiving the ciphertext, the user decrypts it according to his own key. The decryption algorithm is as follows.

$Dec(PK, ct, sk) \rightarrow M$ . Input public parameters  $PK$ , ciphertext  $ct$ , and user's secret key  $sk$ . If the DU meets the access control policy  $A$ ,  $I \subset \{1, \dots, n\}$ ,  $I = \{i : \rho(i) \in A\}$ , compute a set of constants  $\{\omega_i \in R_{q_1}\}_{i \in I}$  with a linear reconstruction algorithm of LSSS, then  $\sum_{i \in I} \delta_i' \omega_i = s_1$ ; compute  $M' = C_0 - K_0 \sum_{i \in I} C_i \omega_i K_i$ ,  $M = M' \bmod p$ ; the DU can download the data through the address  $L$  and decrypt it with the symmetric key  $k$  to obtain data.

The correctness of the successful decryption of the scheme is explained as follows.

$$\begin{aligned}
M' &= C_0 - K_0 \sum_{i \in I} C_i \omega_i K_i = C_0 - K_0 \sum_{i \in I} (arPK_i \delta_i + pe_i') \omega_i K_i \\
&= C_0 - K_0 ar_1 s \sum_{i \in I} (PK_i \cdot K_i) - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) \\
&= C_0 - K_0 ar_1 s \sum_{i \in I} ((SK_i + pe_i) (SK_i^{-1} t + pe_i'')) \\
&\quad - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) = C_0 - K_0 ar_1 s_1 \sum_{i \in I} \\
&\quad \cdot (t + SK_i pe_i'' + pe_i SK_i^{-1} t + p^2 e_i e_i'') \\
&\quad - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) = C_0 - K_0 ar_1 s_1 t - K_0 ar_1 s_1 p \sum_{i \in I} \\
&\quad \cdot (SK_i e_i'' + e_i SK_i^{-1} t + pe_i e_i'') - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) \\
&= PK_0 r_1 s_1 + M + pe' - K_0 ar_1 s_1 t - K_0 ar_1 s_1 p \sum_{i \in I} \\
&\quad \cdot (SK_i e_i'' + e_i SK_i^{-1} t + pe_i e_i'') - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) \\
&= (aSK_0 + pe_0) r_1 s_1 + M + pe' - (SK_0 t^{-1} + pe'') ar_1 s_1 t \\
&\quad - K_0 ar_1 s_1 p \sum_{i \in I} (SK_i e_i'' + e_i SK_i^{-1} t + pe_i e_i'') \\
&\quad - K_0 p \sum_{i \in I} (e_i' \omega_i K_i) = pe_0 r_1 s_1 + M + pe' - pe'' ar_1 s_1 t \\
&\quad - K_0 ar_1 s_1 p \sum_{i \in I} (SK_i e_i'' + e_i SK_i^{-1} t + pe_i e_i'') \\
&\quad - K_0 p \sum_{i \in I} (e_i' \omega_i K_i).
\end{aligned} \tag{2}$$



TABLE 2: Blockchain data structure in search phase.

Identification	From	To	Action	Timestamp	Signature	Transaction
ID_2	DU	BC	Calculation	Timestamp2	Sig2	$T_X, T_Y$
ID_3	BC	DU	Publish	Timestamp3	Sig3	$T_X, T_Y$

TABLE 3: Implementation process of contract.

Input	Search contract User ID, trapdoor, DU address
Output	Ciphertext set ( $ct$ .row) or $\perp$
1.	While (true) do
2.	Calculate $J$ by executing ..
3.	If ( $J == 0$ ) then
4.	Add $ct$ to $ct$ .row
5.	Else
6.	Return ( $\perp$ )
7.	End if
8.	Continue
9.	End while

Then,  $M = M' \bmod p$ , and in order to ensure the correctness of the scheme, the noise term in the scheme must be small enough compared to the ratio of  $q$  to  $p$ .

After successful decryption, the user obtains the data address and the symmetric key and decrypts the data with the symmetric key after obtaining the data from IPFS to obtain the original data.

## 4. Analysis

**4.1. Security Analysis.** This section will discuss the security of the scheme from four aspects according to the security definition in Section 3.3.

### (1) Analysis of IND-CPA security

**Theorem 9.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary  $\mathcal{A}$ , with the advantage  $\epsilon$  to win the game in Definition 6, then there exists a PPT simulator  $\mathcal{B}$  which can decide Decision R-LWE $_{d,q,\chi}$  Problem with advantage  $\epsilon/2$ .*

*Proof.* The Decision R-LWE $_{d,q,\chi}$  Problem is to determine whether the oracle  $O$  is a noisy pseudorandom  $O_s$  or a truly random  $O_s'$ , then the simulator  $\mathcal{B}$  differentiates  $O$  by adversary  $\mathcal{A}$ . First,  $\mathcal{B}$  queries the oracle and receives  $(t+1)$  samples  $(\omega_k, v_k) \in R_q \times R_q$ , where  $k \in \{0, 1, 2, \dots, t\}$ , then proceed as follows.  $\square$

*Initialization phase.* Given a set of attributes  $U$ , the adversary  $\mathcal{A}$  selects an access structure  $A^*$  that wishes to be challenged and sends it to  $\mathcal{B}$ .

*Setup.*  $\mathcal{B}$  runs  $Setup(\lambda, U) \rightarrow PP, MSK$ , let  $PK_0 = p\omega_0 \in R_q$ , and select a pair of uniformly random  $(SK_i, SK_i^{-1}) \leftarrow R_q$  for each attribute in  $U$ . Let  $PK_i = p\omega_i \in R_q$  if  $i \in A^*$ ; otherwise, let  $PK_i = SK_i + pe_i \in R_q$ . Then,  $\mathcal{B}$  sends  $PK = \{a, PK_0, \{PK_i\}_{i=1}^n\}$  to  $\mathcal{A}$ .

*Inquiry Phase 1.*  $\mathcal{A}$  sends private key queries for  $D^* = \{D_1^*, D_2^*, \dots, D_j^*\}$ , where  $D^*$  does not meet the access policy  $A^*$ .  $\mathcal{B}$  runs  $KeyGen$ , computes  $K_0 = SK_0 t^{-1} + pe'' \in R_q$ ,  $K_i = SK_i^{-1} t + pe_i'' \in R_q, \forall i \in D^*$ , and sends  $K = (K_0, K_i)$  to  $\mathcal{A}$ .

*Challenge.*  $\mathcal{A}$  chooses two messages  $M_0, M_1 \in \{0, 1\}$  and sends them to simulator  $\mathcal{B}$ , then  $\mathcal{B}$  randomly selects  $b \in \{0, 1\}$ , if  $b = 0$ ,  $\mathcal{B}$  randomly chooses  $x \leftarrow R_q$  and lets  $C_0 = px_0 \in R_q$ ,  $C_j = px_j \in R_q$ ; if  $b = 1$ , let  $C_0 = pv_0 + M \in R_q$ ,  $C_j = pv_j \in R_q$  for  $j \in A^*$ .

*Inquiry Phase 2.*  $\mathcal{A}$  asks for the key as in phase 1.

*Guess.* Adversary  $\mathcal{A}$  outputs his guess  $b'$  about  $b$  to  $\mathcal{B}$ . If  $b' = b$ , output  $O' = O_s$ ; otherwise, output  $O' = O_s'$ . The advantage of  $\mathcal{A}$  in this game is defined as  $\text{adv}^{\mathcal{A}} = \Pr[b' = b] - (1/2)$ , so the oracle  $O$  has the following two cases.

$O$  is a noisy pseudorandom  $O_s$ . The advantage of  $\mathcal{A}$  is  $\epsilon$ , then  $|\Pr[b' = b | O = O_s]| = (1/2) + \epsilon$  and  $|\Pr[O' = O | O = O_s]| = (1/2) + \epsilon$ .

$O$  is a truly random  $O_s'$ .  $\mathcal{A}$  has no advantage  $\epsilon$  and unable to get information about  $b$ , then  $|\Pr[b' \neq b | O = O_s']| = (1/2)$  and  $|\Pr[O' = O | O = O_s']| = (1/2)$ .

Then, the advantage of simulator  $\mathcal{B}$  is as follows.

$$\frac{1}{2} |\Pr[O' = O | O = O_s]| + \frac{1}{2} |\Pr[O' = O | O = O_s']| - \frac{1}{2} = \frac{1}{2} \left( \frac{1}{2} + \epsilon \right) + \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} = \frac{\epsilon}{2}. \quad (3)$$

Hence, Theorem 9 is proved, and this means that the scheme meets IND-CPA security.

### (2) Analysis of anticollusion attack security

The private key generated by the authority to the user contains the randomly selected reciprocal element  $(t, t^{-1}) \leftarrow R_{q_1}$ , which ensures the uniqueness of the key. At the same time, from the assumption of learning with error, it is difficult for malicious users to restore effective parameter information from their own key. Even if the attributes of colluding users are combined to contain the attributes of the target they want to attack, it is difficult to generate an effective new private key by effective means.

### (3) Analysis of IND-CKA security

**Theorem 10.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary  $\mathcal{A}_3$ , with the advantage  $\varepsilon$  to win the game in Definition 7, then there exists a PPT simulator  $\mathcal{B}_3$  which can decide Decision R-LWE $_{d,q,\chi}$  Problem with advantage  $\varepsilon/2$ .*

*Proof.* The Decision R-LWE $_{d,q,\chi}$  Problem is to determine whether the oracle  $O$  is a noisy pseudorandom  $O_s$  or a truly random  $O_s'$ , then the simulator  $\mathcal{B}_3$  differentiates  $O$  by adversary  $\mathcal{A}_3$ . First,  $\mathcal{B}_3$  queries the oracle and receives  $(t+1)$  samples  $(\omega_k, v_k) \in R_q \times R_q$ , where  $k \in \{0, 1, 2, \dots, t\}$ , then proceed as follows.  $\square$

*Initialization phase.* Given a set of keywords  $W$ , the adversary  $\mathcal{A}_3$  selects  $D_{w_0}^*$  and  $D_{w_1}^*$  as two keywords with the same length that wishes to be challenged and sends them to  $\mathcal{B}_3$ .

*Setup.*  $\mathcal{B}_3$  runs  $Setup(\lambda_2, W) \rightarrow PP, MK$ , let  $sk_0 = p\omega_0 \in R_{q_2}$ , and select a pair of uniformly random  $(pk_i, pk_i^{-1}) \leftarrow R_{q_2}$  for each keyword in  $W$ . Let  $sk_i = pk_i + pe_{-i} \in R_{q_2}$ . Then,  $\mathcal{B}_3$  sends  $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i^{-1}\}_{i=1}^n\}$  to  $\mathcal{A}_3$ .

*Inquiry Phase 1.*  $\mathcal{A}_3$  sends index queries for  $D_{wi}^*$ , where the keyword set of the query cannot be the same as the keyword set of the challenge.  $\mathcal{B}_3$  runs  $IGen(PP, D_{wi}) \rightarrow I_{wi}$ , computes  $I_0 = pk_0 t^{-1} + pe_{-i}' \in R_{q_2}$ ,  $I_i = pk_i^{-1} t + pe_{-i}'' \in R_{q_2}$ ,  $\forall i \in D_{wi}^*$ , and sends  $I_{wi} = (I_0, I_i)$  to  $\mathcal{A}_3$ .

*Challenge.*  $\mathcal{B}_3$  randomly selects  $b \in \{0, 1\}$ ; if  $b = 0$ ,  $\mathcal{B}_3$  randomly chooses  $x \leftarrow R_q$  and lets  $I_0 = px_0 \in R_q$ ,  $I_i = px_i \in R_q$ ; if  $b = 1$ , let  $I_0 = pk_0 t^{-1} + pe_{-i}' \in R_{q_2}$ , and  $I_i = pk_i^{-1} t + pe_{-i}'' \in R_{q_2}$ , then send  $I_{wb} = (I_0, I_i)$  to  $\mathcal{A}_3$ .

*Inquiry Phase 2.*  $\mathcal{A}_3$  asks for the index as in phase 1.

*Guess.* Adversary  $\mathcal{A}_3$  outputs his guess  $b'$  about  $b$  to  $\mathcal{B}_3$ . If  $b' = b$ , output  $O' = O_s$ ; otherwise, output  $O' = O_s'$ . The advantage of  $\mathcal{A}_3$  in this game is defined as  $\text{adv}^{\mathcal{A}_3} = \Pr[b' = b] - (1/2)$ , so the oracle  $O$  has the following two cases.

$O$  is a noisy pseudorandom  $O_s$ . The advantage of  $\mathcal{A}_3$  is  $\varepsilon$ , then  $|\Pr[b' = b | O = O_s]| = (1/2) + \varepsilon$ , and  $|\Pr[O' = O | O = O_s]| = (1/2) + \varepsilon$ .

$O$  is a truly random  $O_s'$ .  $\mathcal{A}_3$  has no advantage  $\varepsilon$  and unable to get information about  $b$ , then  $|\Pr[b' \neq b | O = O_s']| = (1/2)$ , and  $|\Pr[O' = O | O = O_s']| = (1/2)$ .

Then, the advantage of simulator  $\mathcal{B}_3$  is as follows.

$$\frac{1}{2} |\Pr[O' = O | O = O_s]| + \frac{1}{2} |\Pr[O' = O | O = O_s']| - \frac{1}{2} = \frac{1}{2} \left( \frac{1}{2} + \varepsilon \right) + \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} = \frac{\varepsilon}{2}. \quad (4)$$

Hence, Theorem 10 is proved, and this means that the scheme meets IND-CKA security.

#### (4) Analysis of IND-IKGA security

**Theorem 11.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary  $\mathcal{A}_4$ , with the advantage  $\varepsilon$  to win*

*the game in Definition 8, then there exists a PPT simulator  $\mathcal{B}_4$  which can decide Decision R-LWE $_{d,q,\chi}$  Problem with advantage  $\varepsilon/2$ .*

*Proof.* The Decision R-LWE $_{d,q,\chi}$  Problem is to determine whether the oracle  $O$  is a noisy pseudorandom  $O_s$  or a truly random  $O_s'$ , then the simulator  $\mathcal{B}_4$  differentiates  $O$  by adversary  $\mathcal{A}_4$ . First,  $\mathcal{B}_4$  queries the oracle and receives  $(t+1)$  samples  $(\omega_k, v_k) \in R_q \times R_q$ , where  $k \in \{0, 1, 2, \dots, t\}$ , then proceed as follows.  $\square$

*Initialization phase.* Given a set of keywords  $W$ , the adversary  $\mathcal{A}_4$  selects  $A_{w_0}$  and  $A_{w_1}$  as two keyword policies with the same length that wishes to be challenged and sends them to  $\mathcal{B}_3$ .

*Setup.*  $\mathcal{B}_4$  runs  $Setup(\lambda_2, W) \rightarrow PP, MK$ , let  $sk_0 = p\omega_0 \in R_{q_2}$ , and select a pair of uniformly random  $(pk_i, pk_i^{-1}) \leftarrow R_{q_2}$  for each keyword in  $W$ . Let  $sk_i = pk_i + pe_{-i} \in R_{q_2}$ . Then,  $\mathcal{B}_4$  sends  $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i^{-1}\}_{i=1}^n\}$  to  $\mathcal{A}_4$ .

*Inquiry Phase 1.*  $\mathcal{A}_4$  sends trapdoor queries for  $A_{wi}^*$ , where the keyword set of the query cannot be the same as the keyword set of the challenge.  $\mathcal{B}_4$  runs  $TGen(MK, A_{wi}') \rightarrow T_{wi}'$ , set keyword policy  $A_{wi}^* = (F_w, \rho)$ ,  $F_w \in R_{q_2}^{n \times m}$  with row labels  $\rho(j) \in H$ ,  $\forall j \in [n]$ ,  $H \in A_{w'}$ . Generate a vector  $v = (s_2, r_2, \dots, r_m)$ , where  $r_2, \dots, r_m \leftarrow R_{q_2}$ , and  $s_2 \in R_{q_2}$  is the secret to be shared.  $\delta_i' = F_{wi} \times v \in R_{q_2}$ ,  $i \in (1, n)$ , where  $F_{wi}$  is the vector corresponding to  $i$ th row of  $F_w$ . Then, choose a uniformly random element  $r_1' \leftarrow R_{q_2}$ , and noise terms  $e_{-i}' \leftarrow \chi_2$  computes  $T_0 = sk_0 r_2 s_2 + pe_{-i}' \in R_{q_2}$ ,  $T_i = a_2 sk_i r_2 \delta_i' + pe_{-i}' \in R_{q_2}$ , and sends  $T_{wi} = (T_0, T_i)$  to  $\mathcal{A}_4$ .

*Challenge.*  $\mathcal{B}_4$  randomly selects  $b \in \{0, 1\}$ ; if  $b = 0$ ,  $\mathcal{B}_4$  randomly chooses  $x \leftarrow R_q$  and lets  $T_0 = px_0 \in R_q$ ,  $T_i = px_i \in R_q$ ; if  $b = 1$ , let  $T_0 = pk_0 t^{-1} + pe_{-i}' \in R_{q_2}$ ,  $T_i = pk_i^{-1} t + pe_{-i}'' \in R_{q_2}$ , then send  $T_{wb} = (T_0, T_i)$  to  $\mathcal{A}_4$ .

*Inquiry Phase 2.*  $\mathcal{A}_4$  asks for the trapdoor as in phase 1.

*Guess.* Adversary  $\mathcal{A}_4$  outputs his guess  $b'$  about  $b$  to  $\mathcal{B}_4$ . If  $b' = b$ , output  $O' = O_s$ ; otherwise, output  $O' = O_s'$ . The advantage of  $\mathcal{A}_4$  in this game is defined as  $\text{adv}^{\mathcal{A}_4} = \Pr[b' = b] - (1/2)$ , so the oracle  $O$  has the following two cases.

$O$  is a noisy pseudorandom  $O_s$ . The advantage of  $\mathcal{A}_4$  is  $\varepsilon$ , then  $|\Pr[b' = b | O = O_s]| = (1/2) + \varepsilon$ , and  $|\Pr[O' = O | O = O_s]| = (1/2) + \varepsilon$ .

$O$  is a truly random  $O_s'$ .  $\mathcal{A}_4$  has no advantage  $\varepsilon$  and unable to get information about  $b$ , then  $|\Pr[b' \neq b | O = O_s']| = (1/2)$ , and  $|\Pr[O' = O | O = O_s']| = (1/2)$ .

Then, the advantage of simulator  $\mathcal{B}_4$  is as follows.

$$\frac{1}{2} |\Pr[O' = O | O = O_s]| + \frac{1}{2} |\Pr[O' = O | O = O_s']| - \frac{1}{2} = \frac{1}{2} \left( \frac{1}{2} + \varepsilon \right) + \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} = \frac{\varepsilon}{2}. \quad (5)$$

Hence, Theorem 11 is proved, and this means that the scheme meets IND-IKGA security.

TABLE 4: Comparison with other schemes.

Scheme	Problem	Access structure	Searchable	Keyword	Cross-domain	Antiquantum attack	Blockchain
[20]	DBDH	Access tree	√	1	×	×	√
[26]	SDP	LSSS	√	1	×	×	×
[27]	DBDH	Access tree	×	—	×	×	√
[28]	DBDH	Access tree	×	—	×	×	×
[29]	DBDH	LSSS	√	1	×	×	√
[30]	LWE	LSSS	√	1	×	√	×
[31]	LWE	AND	√	1	×	√	√
Ours	RLWE	LSSS	√	n	√	√	√

TABLE 5: Storage cost.

Scheme	Public key	Master key	Private key	Index	Trapdoor
[30]	$(nm + m^2N + 2n + m^2) \log q$	$m^2 \log q$	$2nm \log q$	$2nmA_w \log q$	$2m \log q$
[31]	$(nm + m^2N + n + m^2) \log q$	$m^2 \log q$	$2nm \log q$	$(m + 1)l \log q$	$m \log q$
Ours	$n(N_1 + 2N_2 + 3) \log q$	$n(2N_1 + N_2 + 3) \log q$	$n(A_u + 1) \log q$	$n(A_w + 1) \log q$	$n(A_{w'} + 1) \log q$

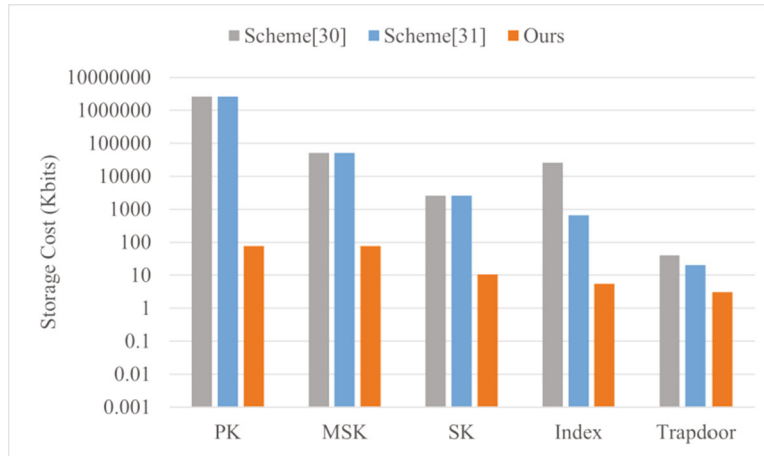


FIGURE 4: The comparison of storage cost.

TABLE 6: Calculation cost.

Scheme	Cost of index generation	Cost of trapdoor generation	Cost of single matching
[30]	$2mnmul$	$2mnmul$	$(4m + 2)mul$
[31]	$l(m + 1)mul$	$2mmul$	$mlmul$
Ours	$(A_w + 1)mul$	$(3A_{w'} + 2)mul$	$(2A_d + 1)mul + mod$

**4.2. Performance Analysis.** Since the scheme in this paper is mainly constructed on the basis of CP-ABE, some schemes based on attribute-based encryption are selected for comparison, including searchable schemes, schemes combined with blockchain and lattice schemes. These schemes are selected to compare their functions, the cost of storage, calculation, and communication.

- (1) Different attribute-based encryption schemes are selected for function comparison. The results are shown in Table 4

Scheme [20] uses searchable encryption technology to realize the search of a single keyword on the blockchain and implements access control according to CP-ABE. The scheme is constructed by bilinear pairing, which has great application prospects in social networks and medical information fields, but the scheme cannot resist quantum attacks.

Scheme [26] pays attention to the problems of high computing cost and low efficiency of searching data in ABE scheme, reduces the local computing cost of users by using outsourcing technology, and proves that the scheme meets adaptive security. However, the scheme only supports single keyword search and cannot resist quantum attacks.

For medical data protection, scheme [27] combines ABE and blockchain technology to enable data to be shared

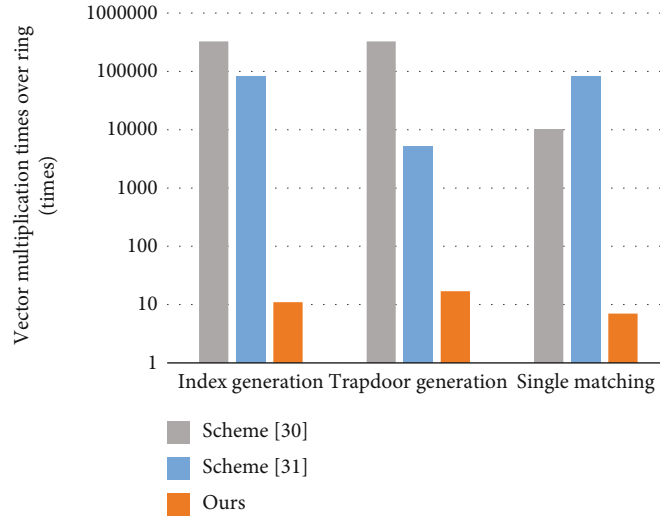


FIGURE 5: The comparison of calculation cost.

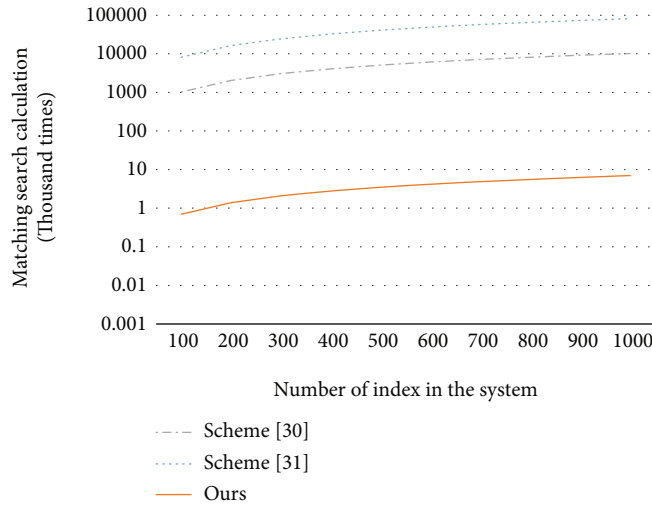


FIGURE 6: The calculation cost for indexes ranging from 100 to 1000.

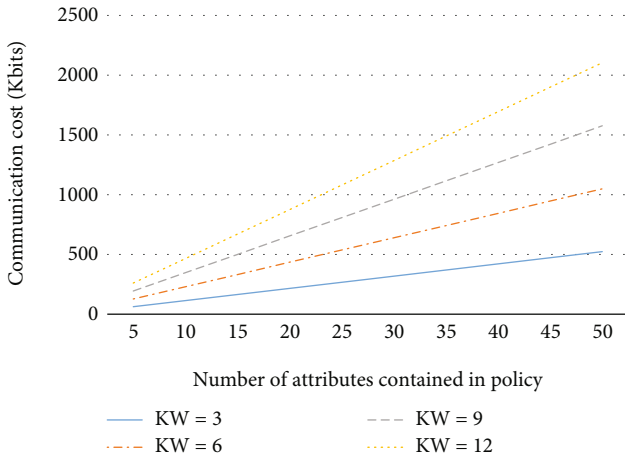


FIGURE 7: The communication cost for indexes ranging.

efficiently and safely among patients, hospitals, and other entities. The scheme does not support ciphertext search and cannot resist quantum attacks.

In the Internet of Things environment, scheme [28] outsources the decryption operation in content encryption to fog nodes, solves the problem that computing is difficult due to the limited resources of Internet of Things devices and also protects users' privacy by constructing false attributes. The scheme does not support ciphertext search and quantum attack resistance.

Scheme [29] combines blockchain and ABE to realize data sharing. The scheme realizes decentralization and avoids the risk of privacy disclosure by third parties and supports ciphertext search. However, the scheme cannot resist quantum attacks.

Scheme [30] solves the problem of ciphertext search in the cloud environment. The scheme only supports single-

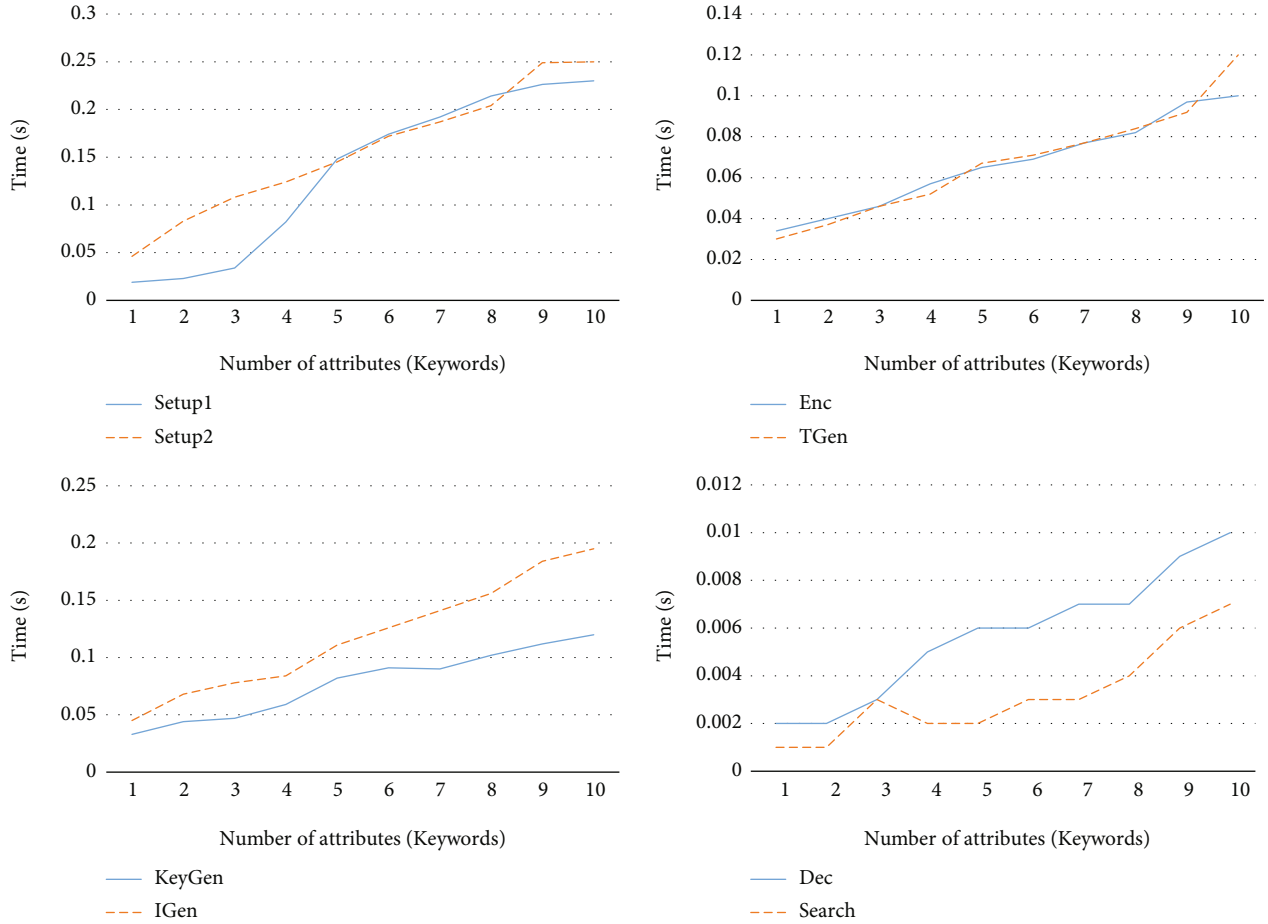


FIGURE 8: Relationship between the number of attributes or keywords and the running time of algorithm.

keyword search. The biggest feature is that it is based on LWE construction and can resist quantum attacks.

Scheme [31] relies on the technical characteristics of Ethereum to solve the problem of single point of failure in traditional systems, and can resist quantum attacks based on the LWE problem. At the same time, the scheme enables data users to generate private keys for visitors, avoiding key abuse caused by third parties. Due to the use of AND gate multivalued access strategy, its expression ability is slightly weak.

Based on RLWE, our scheme can resist quantum attacks, and LSSS has rich access structure, and the scheme realizes cross-domain access based on blockchain and can realize search of multiple keywords.

- (2) Since schemes [20, 26–29] are not based on lattice structure, it is mainly compared with scheme [30, 31]. In schemes [30, 31],  $m$  and  $n$  are the parameters from lattice,  $m \geq 5n \log q$ , and  $l$  is the security parameter in the keyword matching algorithm in scheme [30].  $N_1$  represents the number of all attributes in the system,  $N_2$  represents the number of all keywords in the system,  $A_u$  represents the number of attributes in the user attribute set, and  $A_w$  rep-

resents the number of keywords in index,  $A_{w'}$  indicates the number of keywords in the trapdoor. The results are shown in Table 5; this scheme is superior to scheme [30, 31] in size of system public key, master key, user private key, index, and trapdoor. As shown in Figure 4, the three schemes are analyzed by numerical simulation for visual representation, where  $n = 64$ ,  $q = 129$ ,  $N_1 = 50$ ,  $N_2 = 50$ ,  $m = 5n \log q$ ,  $A_u = 20$ ,  $A_w = 10$ ,  $A_{w'} = 5, l = 32$

- (3) The comparison results with schemes [30, 31] in terms of computational cost are shown in Table 6. Since the cost of addition operation is small, it is not included in the analysis here.  $A_d$  represents the number of keywords required for successful matching, the mul represents the multiplication between vectors on the ring, and the mod represents modular operation, and other parameters have the same meaning as (2). For visual representation, numerical simulation analysis is carried out, the calculation amount of index generation, trapdoor generation, and single matching is shown in Figure 5. When

the index's number in the system is 100-1000, the search matching overhead is as shown in Figure 6

- (4) The main objects of communication cost include ciphertext and index. The encrypted information in the ciphertext is mainly the address of data storage returned by IPFS and the symmetric key used for symmetric encryption of the original data. Set the sum of the two elements as 1280 bit; the index mainly includes the keyword combination of data. Now, simulate the ciphertext overhead of the number of attributes included in the attribute strategy from 5 to 50 when the keywords are 3, 6, 9, and 12, respectively. The results are shown in Figure 7

- (5) Experimental analysis

In order to further analyze the performance of the scheme, we tested 8 algorithms in the scheme. Because there are few simulation experiments related to the lattice attribute-based encryption scheme, it is difficult to effectively compare and analyze with other schemes. Here, the efficiency of the algorithm is mainly tested and analyzed. The experimental environment is AMD ryzen 7-5800H processor 3.20 GHz, 16.0 GB memory, 64-bit Windows 11 operating system. The experimental program is written in c++ language and implemented in QT creator development environment based on NTL library. In this experiment, setting parameters  $q = 8380417$ ,  $p = 3$ , mainly test the running time of each algorithm when the number of attributes or keywords is from 1 to 10. Since the search part of this scheme is similar to the attribute-based encryption system, two algorithms with similar principles are put into the same diagram for analysis. It can be seen from Figure 8 that the running time of the algorithm is proportional to the number of attributes or keywords contained in the algorithm process, and the experimental results are consistent with the theoretical analysis results.

## 5. Conclusion

In this paper, a searchable attribute-based encryption scheme supporting cross-domain access is constructed based on the RLWE. The whole process can be traced based on the blockchain, and the combined search of multiple keywords is supported at the same time. Through analysis, the scheme meets trapdoor search security, anticollusion attack, and the indistinguishability under chosen-plaintext attack. Compared with other schemes, it has certain advantages in function and performance, but the scheme does not consider the change of user attributes. The next step will study the security and efficiency of attribute revocation and update on this basis.

## Data Availability

All data used during the study are available from the corresponding author upon request.

## Conflicts of Interest

The authors state that there is no conflict of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61572521), Engineering University of the PAP Innovation Team Science Foundation (no. KYTD201805), and Natural Science Basic Research Plan in Shanxi Province of China (2021JM252).

## References

- [1] C. Wang, J. Z. Chen, Y. J. Liu, and A. Li, "A cross-domain access control method for large organizations," in *Proceedings of 2014 International Conference on Advances in Materials Science and Information Technologies in Industry (AMSITI 2014)*, pp. 28–33, Xi'an, China, 2014.
- [2] X. H. Yang and H. Wang, "A cross-domain access control model based on trust measurement," *Wuhan University Journal of Natural Sciences*, vol. 21, no. 1, pp. 21–28, 2016.
- [3] S. Shuang and S. D. Chen, "Trusted and efficient cross-domain access control system based on blockchain," *Scientific Programming*, vol. 2020, Article ID 8832568, 13 pages, 2020.
- [4] L. Bai, K. Fan, Y. Bai, X. Cheng, H. Li, and Y. Yang, "Cross-domain access control based on trusted third-party and attribute mapping center," *Journal of Systems Architecture*, vol. 116, no. 5, article 101957, 2021.
- [5] I. Ullah, S. Zeadally, N. U. Amin, M. A. Khan, and H. Khattak, "Lightweight and provable secure cross-domain access control scheme for the Internet of Things (IoT) based wireless body area networks (WBAN)," *Microprocessors and Microsystems*, vol. 81, no. 2, article 103477, 2021.
- [6] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *EUROCRYPT*, pp. 457–473, 2005.
- [7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Alexandria, Virginia, USA, 2006.
- [8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, CA, USA, 2007.
- [9] L. Cheung, J. A. Cooley, R. Khazan, and C. Newport, *Collusion-Resistant Group Key Management Using Attribute-Based Encryption*, Cryptology ePrint Archive Report 2007/161, 2007.
- [10] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography – PKC 2011*, pp. 53–70, Springer, 2011.
- [11] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee, "Functional encryption for threshold functions (or fuzzy IBE) from lattices," in *Public Key Cryptography – PKC 2012*, pp. 280–297, Springer, 2012.

- [12] Y. T. Wang, "Lattice ciphertext policy attribute-based encryption in the standard model," *International Journal of Network Security*, vol. 16, no. 6, pp. 444–451, 2014.
- [13] S. Tan and S. Azmasn, "Lattice ciphertext-policy attribute-based encryption from ring-LWE," in *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, pp. 258–262, Langkawi, Malaysia, 2015.
- [14] G. Kaiyang, H. Yiliang, L. Kai, and W. Riming, "Traceable attribute-based encryption on OBDD access structure from lattice," in *2022 11th International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 210–215, Singapore, Singapore, 2022.
- [15] S. S. Basu and S. Tripathy, "Securing multicast group communication in IoT-enabled systems," *IETE Technical Review*, vol. 36, no. 1, pp. 83–93, 2019.
- [16] L. S. Yao and S. P. Wang, "Attribute-based encryption with equality test in the internet of things," *Microelectronics & Computer*, vol. 36, no. 6, pp. 64–69, 2019.
- [17] P. S. Challagidad and M. N. Birje, "Efficient multi-authority access control using attribute-based encryption in cloud storage," *Procedia Computer Science*, vol. 167, no. 1, pp. 840–849, 2020.
- [18] Y. L. Tian, K. D. Yang, and Z. Wang, "Algorithm of blockchain data provenance based on ABE," *Journal on Communications*, vol. 40, no. 11, pp. 101–111, 2019.
- [19] M. M. Sandoval, H. M. Marin-Castro, and J. L. González, "Attribute-based encryption approach for storage sharing and retrieval of encrypted data in the cloud," *IEEE Access*, vol. 8, pp. 170101–170116, 2020.
- [20] S. F. Niu, Y. Y. Xie, P. P. Yang, and X. Du, "Cloud-assisted attribute-based searchable encryption scheme on blockchain," *Journal of Computer Research and Development*, vol. 58, no. 4, pp. 811–821, 2021.
- [21] X. D. Zhang, T. W. Chen, Y. M. Yu et al., "Model of blockchain data sharing based on ABE," *Application Research of Computers*, vol. 38, no. 8, pp. 2278–2283, 2021.
- [22] S. F. Niu, M. Song, L. Z. Fang et al., "Cloud storage data sharing based on attribute encryption in smart healthcare," *Journal of Electronics & Information Technology*, vol. 44, no. 1, pp. 107–117, 2022.
- [23] P. Kanimozhi and T. Victoire, "Secure sharing of IOT data in cloud environment using attribute based encryption," *Journal of Circuits, Systems and Computers*, vol. 30, no. 6, article 2150102, 2021.
- [24] X. Li and M. Tan, "Electronic certificate sharing scheme with searchable attribute-based encryption on blockchain," *Journal of Physics Conference Series*, vol. 1757, no. 1, article 12161, 2021.
- [25] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," *EUROCRYPT*, pp. 35–54, 2013.
- [26] L. F. Guo and Q. L. Wang, "Adaptive secure outsourced attribute-based encryption scheme with keyword search," *Journal of Computer Applications*, vol. 41, no. 11, pp. 3266–3273, 2020.
- [27] S. Pournaghi, B. Majid, and F. Yaghoub, "MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020.
- [28] O. Nouali, A. Abdelouahab, and S. Ahmed, "SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and fog computing," *Cluster Computing*, vol. 25, no. 1, pp. 167–185, 2022.
- [29] 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 Communications and Mobile Computing*, vol. 2021, Article ID 6658920, 20 pages, 2021.
- [30] U. S. Varri, S. K. Pasupuleti, and K. V. Kadambari, "CP-ABSEL: ciphertext-policy attribute-based searchable encryption from lattice in cloud storage," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1290–1302, 2021.
- [31] X. Wang and Y. L. Chen, "Attribute-based searchable encryption scheme from lattices on Ethereum," *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 33, no. 4, pp. 67–682, 2021.