Keyword Searchable Encryption Scheme based on Blockchain in Cloud Environment

Yanyan Yang 1,*, Mingsheng Hu 1, Yage Cheng 1, Xinxin Liu 1, and Wei Ma 2

1 School of information science and technology, Zhengzhou Normal University, Zhengzhou 450044, China
2 College of information engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

Abstract—With the development of cloud computing, a large amount of sensitive data information is often stored on cloud servers in the form of ciphertext, thereby alleviating local storage pressure while ensuring data privacy. Although encryption helps protect the confidentiality of the data, a secure and efficient search of ciphertext data is a challenging issue. Aiming at the problems of searchable encryption schemes in the existing cloud environment, such as low search efficiency and accuracy, and poor security, this paper proposes a traceable, keyword searchable encryption scheme based on blockchain technology. This solution makes use of the advantages of the non-tampering and non-repudiation of the blockchain to store query records on the blockchain, thereby achieving traceability of user queries and preventing illegal users from leaking sensitive data. Finally, under the random oracle model, the scheme is proved to be adaptive and indistinguishable. Through comparative analysis, it is shown that the scheme has high search accuracy and security.

Keywords—cloud Server, searchable encryption, blockchain.

I. INTRODUCTION

With the rapid development of cloud computing, more and more users upload data to cloud server storage in order to better store and share data. However, the cloud server generally belongs to a third party, so the access and use of data is not controlled by the user, which may cause the user's data to be leaked. In order to protect the confidentiality of the data, the data is generally encrypted and then uploaded to the cloud server. Although encryption technology protects the privacy and integrity of data, it is not conducive to data retrieval [1]. If you want to retrieve files containing certain keywords from the encrypted files stored on the cloud server, according to traditional practices, users need to retrieve the encrypted files from the cloud server first, decrypt them and then retrieve them. However, if a large number of encrypted files are stored on the cloud server, this will bring great communication overhead. In addition, decrypting all retrieved encrypted files will cause huge computational overhead to the user, and the efficiency of the user to obtain the required files is extremely low. Obviously, this method will bring huge computing and storage overhead to users, and to a certain extent lose the advantages brought by cloud storage services [2].

By establishing a secure index for data outsourced to cloud servers, searchable encryption technology makes it possible to retrieve ciphertext data, avoiding the problems of traditional practices. In recent years, public key searchable encryption technology has attracted much attention because it allows users to perform keyword searches without decrypting the ciphertext. In traditional public key searchable encryption, the sender uses the receiver's public key to generate a searchable ciphertext and uploads it to the cloud server. The receiver uses his own private key and keywords to generate a trapdoor and uploads it to the cloud server. The cloud server will use the received trapdoor to retrieve whether the ciphertext contains the keywords in the trapdoor, and finally return the qualified ciphertext to the data receiver [3]. In 2000, Song et al. [4] gave the concept of searchable encryption for the first time. The data owner establishes a secure index of keywords and stores it with the data ciphertext to the cloud server platform. The ciphertext data of the word, thus achieving the direct operation of the ciphertext data, but the symmetric key is used to encrypt the data, and a special secure channel is required to transmit the symmetric key. In 2004, Boneh et al. [5] first proposed Public Key Encryption with Keyword Search (PEKS). In this scheme, multiple data senders use the public key of a data receiver to encrypt data without the need for a special secure channel to transmit the key. The security is proved under the random oracle model, which is suitable for the multi-data provider-single user scenario. In order to be able to implement a more practical search scheme, based on the single-keyword search function expansion, literature [6-7] proposed an encryption scheme that supports multi-keyword search. In order to meet the search needs of different users, various schemes have been proposed, such as a search scheme based on attribute encryption [8-9], a search encryption scheme based on fuzzy keywords [10], and so on. In view of the characteristics of blockchain technology such as tamper resistance, public verification, and decentralization [11]. In 2018, literature [12] first proposed a single-keyword searchable encryption scheme based on blockchain. Subsequently, literature [13-14] also constructed a searchable encryption scheme based on the blockchain, but did not support fine-grained user access.

Aiming at the problems of search efficiency and accuracy with low search efficiency and poor security in the existing cloud environment, this paper proposes a searchable encryption scheme with traceable keywords. This solution makes use of the advantages of the non-tampering and non-repudiation of the blockchain to store query records on the blockchain, thereby achieving traceability of user queries and preventing illegal users from leaking sensitive data. In addition, the vector model is used to build indexes and query traps to achieve more flexible and complex multi-keyword queries. Finally, under the random oracle model, the scheme is proved to be safe and efficient.
II. RELATED DEFINITIONS

A. Bilinear Mapping
Suppose $p$ is a large prime number, $Z_p$ is a finite field, $G$ and $G_r$ are two cyclic groups of order $p$, and $g, g_r$ are generators corresponding to $G, G_r$, respectively. Bilinear mapping $e : G \times G \rightarrow G_r$ can be defined [4]:

1) Bilinearity: For any $a, b \in Z_p$ , there is $e(g^a, g^b) = e(g, g)^{ab}$;

2) Non-degenerate: $e(g, g) \neq 1$;

3) Computability: For any element $g, g_r \in G, G_r$, $e(g, g_r)$ can be calculated efficiently.

In group $G$, as the security assumptions of cryptography, there are mainly the following cryptographic difficulties.

B. Calculate the Diffie-Hellman problem (CDH)
Given a triple $(g, g^a, g^b) \in G$ is unknown, and $g$ is a generator of the cyclic group $G$. Determine whether $g^{ab}$ is true [11].

C. Determining the Diffie-Hellman problem (DDH)
Given a quaternion $(g, g^a, g^b, g^c) \in Z_p$ is unknown, and $g$ is a generator of the cyclic group $G$, determine whether $g^{abc}$ [1] holds [1].

D. Bilinear Diffie-Hellman problem (BDH)
Given a quaternion $(g, g^a, g^b, g^c) \in Z_p$ is unknown and $g$ is a generator of the cyclic group $G$, calculate $e(g, g)^{ab}$ [1].

E. Blockchain
Blockchain [15] is a typical distributed ledger technology that supports data verification, sharing, calculation, storage and other functions through multilateral autonomous technical means such as consensus. In different application scenarios, blockchain can store and process different data.

From the perspective of the organizational structure and operating principles of the blockchain, the blockchain can be viewed in a narrow sense as a one-way chain data structure with blocks as the unit and connected in a chronological order. Through the consensus mechanism and cryptography Technologies such as component and system fault tolerance ensure the consistency and security of data shared by nodes in a distributed network. From an application point of view, blockchain is a composite distributed network technology that integrates cryptographic algorithms, distributed networks, consensus mechanisms, game theory and other technologies. It uses a chain-type block structure to store data and uses a consensus mechanism to implement transactions. The update and sharing of the system, the use of cryptography technology to ensure the security of transactions, the use of automated script codes to achieve programmability and autonomy, and the use of economic incentive mechanisms to stimulate nodes to independently maintain system stability, forming a new and autonomous distributed infrastructure and computing paradigm.

F. Public key Encryption with Keyword Search (PEKS)
PEKS [16] can be traced back to the problem of untrusted server routing, namely Bob wants to send mail to Alice through the mail server. In order to ensure the privacy of the mail, the mail server needs to correctly send the mail to Alice according to the content of the mail without knowing the content of the mail.

Boneh et al. [5] applied searchable encryption technology to asymmetric cryptography for the first time, and proposed the concept of PEKS. The algorithm is described as follows:

KeyGen($\lambda$) $\rightarrow$ $(pk, sk)$ : Enter the security parameter $\lambda$, and output a pair of public and private keys $(pk, sk)$. The public key is public and the private key is kept by the user.

$PEKS(pk, w) \rightarrow PEKS : $ Input public key $pk$, keyword $w$, output searchable ciphertext $PEKS$.

$Trapdoor(sk, w) \rightarrow T_w : $ Enter private key $sk$, keyword $w$, and output trapdoor $T_w$.

$Test(pk, PEKS, T_w) \rightarrow (1 \ or \ 0) : $ Enter the public key $pk$, you can search for the ciphertext $PEKS$, the trapdoor $T_w$, if the ciphertext contains the keyword specified in the trapdoor, output 1; otherwise, output 0.

III. PROBLEM DESCRIPTION

A. System Mode
The entities included in this solution are Data Owner (DO), Data User (DU) and Cloud Service (CS), Trusted Third Party (TTP), and Blockchain (BC). The system model is shown in Figure 1.

![System model](image)

Fig. 1. System model.

1) Data Owner: The data owner is responsible for the processing of the original document, and the original document set is encrypted with a symmetric key to generate a ciphertext document set. In addition, in order to realize the retrievability of the ciphertext document collection, it is necessary to extract keywords and generate a security index, and simultaneously upload the keywords and the corresponding security index, and upload the encrypted ciphertext document collection to a cloud server.
2) **Cloud Server**: The cloud server provides data storage services and data sharing services to data owners and data users. The first is that the cloud server receives and stores the data sent by the data owner. The second is that after receiving the search trap of the data user, the cloud server performs the search processing according to the security index and returns the search results that meet the search requirements to the data user. In addition, users can track illegal search operations.

3) **Blockchain**: The blockchain mainly stores the search records of data users, including data users and search traps. When the cloud server proposes an illegal search operation to track users, the blockchain feedbacks the corresponding results according to the stored search records.

4) **Trusted Third Party**: mainly generate system public parameters and master keys.

5) **Data User**: A data user is a legal user authorized by the system, who can legally retrieve the private data of the data owner stored in the cloud server, formulate a search trap according to his own needs, and send data to the cloud server and block. The cloud server submits search traps, and then waits to receive and decrypt search results from the cloud server.

### B. Security Model

This article assumes that the cloud server is semi-honest but curious. It can correctly execute the user's query request in accordance with the requirements of the plan, and will not delete or modify the data uploaded by the data owner. But the cloud server is curious. It may try to obtain additional information from the security index and trapdoor. Therefore, the solution in this article mainly considers the following two security models:

**Known ciphertext model**: The adversary knows the ciphertext information stored by the user, including the encrypted document collection, ciphertext index tree and trapdoor, but does not know the key. The adversary can only conduct known ciphertext attacks based on this information.

**Known background information model**: The cloud server can not only get all the information available in the known ciphertext model, but also has the ability to record query results, analyze the relationship between different trapdoors, and analyze data statistics.

### IV. Specific Scheme

**SE=(SE.KG(λ),SE.Encrypt(K,m),SE.Decrypt(K,m))** is an IND-CPA secure symmetric encryption scheme, and **SC=(SC.KG(λ),SC.Signcrypt(SK,PK, m),SC.Unsigncrypt (VK,SK,C'))** is a PKI signcryption scheme that satisfies confidentiality (IND-CCA2) and unforgeability (EUF-CMA). Here, \( \lambda, m, C, K, SK, PK, PK', SK' \) represent the security parameters, message, signed ciphertext, symmetric key, sender's private key, receiver's public key, sender's public key, and receiver's private key, respectively. A multi-keyword searchable encryption scheme based on blockchain is composed of the following five polynomial time algorithms, which are described as follows:

1) **Setup(\( \{\lambda\} \))**: Executed by a third-party trusted center, given the security parameter \( \lambda \) as input, and output system parameter \( \text{params} \).

2) **KeyGen(\( \{\lambda\}, \text{params} \))**: Executed by a third-party trusted center. The algorithm takes \( \lambda \) and \( \text{params} \) as inputs and calls algorithms \( \text{SC.KG} \) and \( \text{SE.KG} \). The sender's public and private key pair \( (PK_s, SK_s) \) and the receiver's public and private key pair \( (PK_r, SK_r) \) and a symmetric key \( k \) are output. In this article, the default cloud server is semi-honest.

3) **Encrypt(\( \text{params}, PK_s, SK_s, w \))**: Executed by the data owner to calculate \( CT = \text{SC.Signcrypt}(PK_s, SK_s, w) \). Finally, the sender uploads the \( CT \) to the cloud server.

4) **Trapdoor(\( \text{params}, K, PK_s, N, w' \))**: is a search trapdoor generation algorithm that is executed by the data user. The algorithm takes \( \text{params}, K, PK_s \), a random number \( N \), and the keyword \( A \) to be searched as input, and outputs a search trap for the keyword \( w' \) of the sender's public key \( T_w = \text{SE.Encrypt}(K, w || PK_s || |N|) \). Then, the receiver sends \( T_w \) to the cloud server, and at the same time saves the search record \( SR = (st, suID, T_w) \) on the blockchain. \( st \) represents the search time, and \( suID \) represents the user ID.

5) **Text(\( \text{params}, T_w, CT \))**: The verification algorithm is executed by the cloud server. Calculate \( w || y' || N = \text{SE.Decrypt}(K, T_w) \) and \( w' = \text{SC.Unsigncrypt}(PK_s, SK_s, CT) \), if \( w' = w' \), output 1; otherwise, output 0.

### V. Security Analysis

**Theorem 1**: Suppose \( SE \) is an IND-CPA secure symmetric encryption scheme, \( SC \) is a PKI-based signcryption scheme, and satisfies the confidentiality (IND-CCA2) and unforgeability (EUF-CMA) under the random prediction model, then The proposed scheme is semantically secure against selected keyword attacks under a random oracle model.

**Proof**: Assuming that in the corresponding security game, there is a polynomial-time adversary \( A \) who can break our scheme, then we can construct an adversary \( \mathcal{E}' \) to break the IND-CCA2 security of the signcryption scheme \( SC \) used in the encryption algorithm. We will prove that \( \mathcal{E}' \)’s advantage over SC's IND-CCA2 game is at least \( \frac{1}{2} - |Pr[\mathcal{A}_{\text{adv}} \times 1/2] | \). In the proof, challenger \( \mathcal{C} \) is used to build \( \mathcal{E}' \), and \( \mathcal{E}' \) uses \( \mathcal{A} \) to achieve its goal, that is, the scheme is semantically secure against the selective keyword attack (SS-CKA).

**Initialization** : At this stage, the challenger \( \mathcal{C} \) first runs the algorithms \( \text{params} \leftarrow \text{Setup}(\{\lambda\}) \) and \( \{PK_s, SK_s\}, \{PK_r, SK_r\}, K \leftarrow \text{KeyGen}(\{\lambda\}, \text{params}) \), and sends the common parameter \( PK_s, PK_r, \text{params} \) to the adversary \( \mathcal{A} \).

**Phase one**: Adversary \( \mathcal{A} \) inquires a polynomial bounded number in an adaptive way:

\( \mathcal{A} \) asks the keyword \( w_i \) about the public key \( PK_s \) and the ciphertext of \( PK_r \). In response, \( \mathcal{C} \) runs the algorithm
Encrypt(\(\text{params}, PK_s, SK_s, w\)) and sends the output \(CT_i\) of the algorithm to \(\mathcal{A}\).

Signcryption query: \(\mathcal{A}\) asks the keywords corresponding to the searchable ciphertext \(CT_i\). At this point, \(C\) runs algorithm \(\text{SCLSigncryption}(PK_s, SK_s, CT_i)\) and sends the output \(w_i\) to \(\mathcal{A}\).

Trapdoor query: \(\mathcal{A}\) can make a trapdoor query to \(w_i\) about any sender's public key \(PK_s\) and \(PK_a\).

Run the algorithm \(\text{Trapdoor}(\text{params}, K, PK_s, N, w)\) and return its result \(T_w\) to \(\mathcal{A}\).

Challenge: First, \(\mathcal{A}\) generates two keywords \(w_0, w_1\) of equal length and a state message \(S\). It is required that in phase one, \(\mathcal{A}\) never makes a ciphertext query or trapdoor query about \(w_0\) or \(w_1\) about \(PK_s\) and \(PK_a\). Then \(\mathcal{A}\) sends \(w_0\) and \(w_1\) to \(C\). \(C\) randomly selects a bit \(b \in \{0,1\}\) and calculates \(CT^* \leftarrow \text{SCSigncryption}(PK_s, SK_s, w_b)\). Finally, \(C\) returns \(CT^*\) to \(\mathcal{A}\) as the challenge ciphertext.

Phase two: As in phase one, \(\mathcal{A}\) can adaptively make a polynomial bounded number of queries. The limitation is that \(\mathcal{A}\) cannot make a ciphertext inquiry or trapdoor inquiry about \(w_0\) or \(w_1\) about \(PK_s\) and \(PK_a\), and has not made a sign decryption inquiry about \(CT^*\).

Guess: \(\mathcal{A}\) outputs a bit \(b^*\). If \(b^* = b\), then \(\mathcal{A}\) wins the game. The advantage is \(|\Pr[b^* = b] - 1/2| \leq \varepsilon\), then the advantage of \(\mathcal{A}\) is negligible, and the plan is proven.

**Theorem 2:** Scheme can ensure that users have non-repudiation and traceability.

**Proof:** The solution uploads each user's search record, that is, \(SR = (st, sudD, w', T_w')\), in the form of a hash value, and the hash value corresponds to the block one-to-one. Therefore, if a user modifies his search record, he will cause the hash value of the block storing the record to change. Similarly, all subsequent blocks connected to the block must be modified at the same time to prevent detachment from the blockchain, but the calculation of the hash value is time-consuming, and it is almost impossible to modify multiple blocks at the same time, so the search records recorded on the blockchain are impossible to tamper with, which guarantees the non-repudiation of users.

In addition, each block in the blockchain contains the IDs of the previous block and the next block. It is a chain structure with a sequence. Therefore, the cloud server can push the chain structure until it reaches Starting node to find relevant user information and ensure that users have traceability.

**VI. CONCLUSION**

This article proposes a keyword searchable encryption scheme based on the blockchain. This scheme not only can resist internal keyword guessing attacks, but also can achieve higher efficiency on the receiver side when generating search trapdoors. In this phase, the scheme only uses a symmetric encryption algorithm. In addition, this solution stores the search traps of various data users on the blockchain, and the blockchain is tamper-resistant and traceable. Therefore, when users are dishonest or have illegal searches, the cloud server can track them. The identity of the user. However, the retrieval efficiency of the blockchain is far inferior to that of cloud servers. In order to improve the data retrieval efficiency and ensure the traceability of data, further research is needed.

**VII. ACKNOWLEDGMENT**

This paper was supported by the National Key Research and Development Project (2016YFB0502300), Henan Key Research Projects of Universities (20A413008, 21B520022), and Science and Technology Program of Henan Province (202102210359).

**REFERENCES**


