International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1098-1107

Using Blockchain Data Query Algorithm to Enhance The Security In Logistics

K. Kusumakumari
Department of Computer Applications
Madanapalle Institute of Technology & Science, India

Mr. S. Bala Murugan
Assistant Professor
Department of Computer Application

Abstract:
In order to ensure the security of logistics information and to query information quickly and efficiently, using searchable encryption algorithms, combined with the characteristics of the blockchain, a searchable and encrypted logistics information blockchain data query algorithm is proposed. First, the logistics information is divided into multiple data files, encrypted with an asymmetric searchable encryption algorithm, and then stored in the cloud server. The keyword index value is extracted from each data file and uploaded to the blockchain. This solution can be used at any time Update and query data. Finally, analyze the correctness, completeness and safety of the scheme of this article, which proves the feasibility of this scheme.
Keywords*— Blockchain, searchable encryption, asymmetric encryption, logistics information, dataquery.*

Date of Submission: 05-06-2022 Date of acceptance: 20-06-2022

I. INTRODUCTION:

In recent years, with the rapid development of e-commerce, the business volume of the logistics industry has shown an explosive growth trend. The rapid changes in the Internet industry have made logistics management more and more informational, and logistics management and operation meth- ods, means and strategies are gradually becoming intelligent. However, behind this trend, there are many problems that need to be solved. There are many areas and long time spansin logistics links, so supervision is difficult, and counter- feiting is difficult to eradicate. In 2008, Satoshi Nakamoto invented Bitcoin [1], and virtual currency became popu-lar all over the world. The blockchain technology in Bit- coin has attracted wide attention from scholars at home and abroad because of its decentralization, tamper resistance, and traceability. Blockchain technology can solve the prob- lem of excessive power in the ‘‘central’’ organization of The associate editor coordinating the review of this manuscript andapproving it for publication was Wenbing Zhao .

traditional logistics enterprises. Through the establishment of a transparent unified information platform by multiple parties, real-time viewing and information transmission are ensured, so as to realize all parts of information in the entire link from production to transportation Can be traced back. Blockchain can accommodate all users in the logistics ser- vice process, In the entire information transmission process, the logistics blockchain is formed through data encryption and consensus verification, thereby ensuring the authenticityand transparency of logistics service transaction informa- tion, and ensuring that the information will not be tampered with and can be queried and traced to the source [2].

Blockchain technology effectively solves the pain points of traditional traceability systems by using its technical fea- tures such as distributed storage, encryption algorithms and time stamps. The types of blockchains can be divided into public chains, consortium chains, and private chains accord-ing to the characteristics of their members. The consortium chain refers to a blockchain network that only allows spe- cific group members and limited third parties to access [3].

Because the consortium chain has an access mechanism and uses a high-performance consensus algorithm, it usually hashigher transaction performance than the public chain. The demand background of the traceability chain matches the consortium chain, that is, the logistics information needs a faster speed And lower cost, so this article chooses consor- tium chain as the basic network of logistics information query system.

The issue of logistics information privacy has always been a major research hotspot in the logistics industry. The main issue is how to encrypt and quickly and accurately query logistics information, Searchable encryption technology caneffectively solve this problem. In 2000, Song *et al.* [4] pro- posed the first searchable

encryption scheme that can realize keyword search on ciphertext. The basic application process of searchable encryption technology is: the user encrypts his own data and uploads it to the remote server, and submits its keyword trapdoor to the server when it needs to retrieve the file, and then the server uses the trapdoor to search for the ciphertext and return it to the user. During the whole process,the server will not obtain any information about the ciphertext and its keywords [5]. Searchable encryption is divided into symmetric searchable encryption and asymmetric search- able encryption according to the encryption type. The asym- metric searchable encryption scheme was first proposed by Boneh *et al.* [6], and the document [7] first proposed the concept of attribute-based searchable encryption. The proxy re-encryption scheme proposed in [8] is the first scheme in which data users can authorize keyword search power to other users. These three searchable encryptions are all asymmetric searchable encryptions. Multi-user scenarios usually use an asymmetric searchable encryption mechanism, that is, use the user's public key to encrypt data, and only users with the corresponding private key can generate search credentials and decrypt the searched ciphertext, which is more secure than symmetric encryption, Is more suitable for logistics informa-tion, but its disadvantage is that the algorithm is complex andthe encryption and decryption speed is slow.

To sum up, this article combines the three major characteristics of blockchain technology: decentralization, non-tamperable data, and traceable data, and proposes a searchable and encrypted logistics information blockchain data query algorithm. In order to solve the problem of the slow encryption and decryption speed of the asymmetric searchable encryption method, the logistics information is first divided into multiple data files, encrypted by the asym- metric searchable encryption algorithm, and then stored in the cloud server. The keyword index value is extracted from eachdata file and uploaded to the blockchain, thereby reducing the burden of encryption and decryption caused by excessive information. The algorithm can update and query data at anytime. In this paper, the encryption and decryption processis explained in detail and a series of sub-algorithms are designed. Finally, the correctness, integrity and security of the algorithm are analyzed to prove the feasibility of the algorithm.

II. WORK RELATED

In order to ensure that the source of the goods is regular, it can be verified by querying the logistics information. How- ever, the logistics information can easily be tampered with and forged. The traditional search method uses the search technology based on plain text, that is, whether the keywordssubmitted by the query user or the data information in the server database are given in plain text, this also causes serious information leakage. Because any malicious server can obtain information such as query keywords and query results ofthe query user, which seriously endangers personal security and privacy. In order to solve this problem, some scholars have proposed a scheme of searchable encryption and query based on ciphertext. In this mode, the basic technology of cryptography is used to ensure users' private information and personal safety [9]. The underlying data storage system of most blockchain systems uses LevelDB [10], which is a data storage system designed for write-intensive applications, at the expense of data read performance in exchange for improved write performance. However, in practical appli- cations, the amount of data written per unit time of the blockchain system is not large. For example, the transaction write volume of the Ethereum system is about 7-10 transac-tions per second, and the current transaction write volume of the Bitcoin system is about 1 transaction per second. The high-speed write advantage of LevelDB cannot be reflected. With the increase of data in the blockchain system and the expansion of applications, frequent queries often need to be processed. The write performance of the underlying stor- age system is excessive but the read performance is insuf- ficient, which has become the main bottleneck that limits query performance. Most of the data storage systems used by blockchain systems are unstructured data storage systems based on the Key-Value model, such as LevelDB. These systems only support insertion and query based on Key val- ues, and do not support relational operations for complex queries. The logistics industry, e-finance, e-commerce and other blockchain applications have an urgent need for rela- tional and analytical queries. Therefore, the query function of the existing blockchain system is extremely limited, and it isdifficult to meet the needs of practicalecond, and the currenttransa applications.

In [10], a method of adding additional indexes to levelDB is proposed to optimize query efficiency. However, this methodwill face write function degradation and query performance is limited by the bottleneck of the blockchain system. In [11],also based levelDB proposed a method for adding an inter- nal index to establish secondary indexes, a query with the key value to improve the efficiency of tracking and tracing block chain. In [12], a practical searchable encryption scheme based on connected keywords is proposed, which avoids memorizing the location of keywords by making an index of keywords. Share files safely and confidentially without introducing a trusted third party. However, in this scheme, multiple keywords use a single trapdoor, which may cause hash collision problems. In [13], the blockchain database is used to store the encrypted form of personal privacy data, but when the blockchain database is constructed, some plain text is still retained to achieve the identification of the data, and it is impossible to completely realize the confidentialityof the data. In [14], a blockchain data privacy protection method based on searchable encryption is proposed. This method uses

the blockchain to store encrypted data and uses the mathematical characteristics of bilinear mapping to construct a blockchain transaction sheet. The encrypted information of keywords is added to the transaction order for keyword search; finally, the user uses the searchable encrypted private key to construct a trapdoor to search for thespecified keyword. In [15], a model suitable for encrypted storage and efficient and secure retrieval of massive data is proposed. There is no data decryption operation in the entire retrieval process, which effectively guarantees the security of data retrieval. However, this scheme directly encrypts a large amount of data, which is too burdensome. In [16],a point-to-point encryption method for power system com- munication data based on blockchain technology is proposed. This method improves the stability of traditional communi- cation data encryption methods, but this method is not suit- able for scenarios with large amounts of data. In [17], [18], in the context of medical treatment, combining blockchain and searchable encryption technology, two schemes that canshare medical records between different hospitals are pro- posed. But the disadvantage is that the two schemes storeall the encrypted data in the blockchain, which imposes a certain storage burden on the blockchain. In [19], in view of the problem that most searchable encryption only supports single-keyword search, a multi-keyword search scheme is proposed, and the scope can be reduced to avoid irrelevant documents, thereby reducing the amount of search calcu- lations. However, this solution does not take into account the problem of data update, and is not suitable for sce- narios where data changes in real time, such as logistics information.

The scale of coverage of the logistics industry is getting larger and larger. This paper analyzes the shortcomings of the current logistics information query algorithm and pro- poses a more secure and more suitable logistics information query algorithm for huge data volumes. The algorithm divides the logistics information into multiple data files, which are encrypted and stored on the cloud server, and the keyword index is stored on the blockchain, which can prevent the cloud server from tampering with the data and reduce the storage pressure of the blockchain, Making the program more scal- able. This paper uses an asymmetric searchable encryption algorithm to encrypt logistics information, and designs the corresponding encryption process. For encrypted informa- tion, keyword query is supported to improve query efficiencyand accuracy. In actual application scenarios, the amount oflogistics information data is huge and updated in real time. Therefore, this paper proposes a scalable query algorithm combined with blockchain technology to ensure the authen-ticity of logistics information.

III. DATA QUERY ALGORITHM

A. SOLUTION DESCRIPTION

Due to the large amount of logistics information data, in order to improve query efficiency, all data D is divided into multiple data files $D_1, D_2, , D_n$. The data file D $D_1, D_2, , D_n$ is stored on the cloud server. In orderto protect the privacy of the data, the data is encryptedbefore uploading to the cloud server. Since cloud serversare not completely credible, it is hoped that there will be amechanism that can remove the trust reliance on the serverand ensure that encrypted data is not tampered with. The emergence of blockchain technology can effectively solve this problem. Blockchain is a distributed ledger that canrestrict unauthorized access while providing data integrityand transparency. Each encrypted document is recorded onthe blockchain after being recognized and certified by theconsensus mechanism. Since it is difficult to search forkeywords on the encrypted data of the blockchain network,before the document is encrypted, a set of keywords W $W_1, W_2, , W_m$ is identified in D, and a secure index table I is generated. Send the inverted index table I to the cloudserver and embed it in the smart contract. Whenever you wantto search for keywords, a trapdoor will be generated, sent tothe cloud server, and further sent to the smart contract. Thesmart contract is automatically executed, searches for the rel-evant content of the keyword, uses the consensus mechanismto verify the incoming search requests and sort the requests.After the search is completed, the results are returned to theclient, the client can download the required data, and thetrapdoor related information is also stored on the blockchainledger.

B. BASIC DEFINITION

The searchable encryption algorithm based on logistics blockchain designed in this paper includes 8 polynomial time algorithms:

$$\Pi = (KeyGen, SigGen, Build\_Index, Adopt, Generate\_Trap, Record, Search\_Outcome, Dec)$$

(1) K, k_s, k_{pr}, k_{pu} $KeyGen\ 1^{\lambda}$: Represents a probabilistic key generation algorithm, λ is the security parameter as input, which returns the master key K. The asymmetric key pair k_{pr}, k_{pu} and the session key k_s are derived from the master key K. The algorithm is executed by the client.

(2) $Sig\ SigGen\ k_{pu}$: Represents the deterministic signature generation algorithm, which requires the client's public key k_{pu} as input, and the MSP module generates the signature Sig, which can be verified by the CA. The algorithm is run by clients and peer nodes that participate in the work of the system.

(3) $(I, Enc_K(D)) \leftarrow Build\_index(K, D)$: The algorithm is a deterministic algorithm run by the client. It takes a collection of master key K and documents D as input and returns a secure index I and encrypted documents $Enc_K(D)$. The index table K is a mapping that can show whether keywords in the document exist.

(4) $Embed\ Adopt(Sig, I)$: This algorithm is a deterministic algorithm triggered by the client. The algorithm takes the signature Sig and index table I as input and embeds it into the smart contract.

(5) $T_w \leftarrow Generate\_Trap(K, k_s, w)$: The probability algorithm run by the client takes the master key K, session key k_s, and keyword w as input and outputs trapdoors T_w.

(6) $Append \leftarrow Record^{Enc_K(D)}_{T_w}$: Represents a deterministic algorithm that stores and records transactions on the blockchain, and encrypted documents $Enc_K(D)$ or trapdoors T_w are attached to the blockchain through a consensus mechanism.

(7) $X \leftarrow Search\_Outcome(k_s, SC, I, T_w)$: A deterministic algorithm run by the cloud server with the help of a smart contract. The algorithm takes an index table I and trapdoor T_w as input and returns I, where I is a set of encrypted document identifiers, denoted as $Enc_K(id(D))$.

(8) $D_i \leftarrow Dec(K, X)$: A deterministic algorithm executed by the client that requires the client master key K and encrypted document identifier $Enc_K(id(D))$ to decrypt and restore the document id to query the corresponding blockchain data segment.

blockchain database, file user (user who queries information). The application process of searchable encryption is shown in Figure 1.

The searchable encryption steps are as follows:

(1) The logistics company uses the key to encrypt the plaintext file and uploads it to the cloud server. At the same time, it uses the searchable encryption key to encrypt the keyword and uploads it to the blockchain database.

(2) During the query process, the user uses the searchable encryption key to encrypt the keywords waiting to be queried to generate a trapdoor. At the same time, the trapdoor does not reveal any information about the keywords, and then sends the encrypted keywords to the blockchain database.

(3) The blockchain database takes the trapdoor as input and executes the matching algorithm to find the index value corresponding to the trapdoor, query the corresponding file in the cloud server according to the index value, and return all ciphertext files that successfully match the index value.

(4) The user receives the ciphertext file and decrypts it with the key.

When encrypting data on the chain, a block chain transaction sheet needs to be constructed. Without changing the original system of the block chain database, a keyword for ciphertext query is added to the blockchain transaction sheet.

The meaning of individual parameters is shown in Table 1.

TABLE 1. Symbols and description.

Parameter	meaning
SC	Smart Contract
MSP	Membership Service
Sig	Client signature
CA	Certification Center
ID	Identity identifier
K	Master key
k_s	Session key
(k_{pr}, k_{pu})	Public key and private key pair
Enc_K	Probabilistic encryption algorithm using master key
Dec_K	Enc_K corresponding decryption algorithm
$H(\cdot)$	Keyed one-way hash function
$\|W\|$	Total number of different keywords identified

C. ALGORITHM DESIGN

Searchable encryption technology is different from query based on plaintext. It is based on encrypted files for query, so it is necessary to provide a ''tag'' similar to plaintext query. The ''tag'' is made by encrypting keywords with searchable encryption public keys. Encrypting keywords makes it impossible to obtain any information about plaintext through encrypted keywords. Therefore, compared to plaintext tags, tags made with searchable encryption technology are safe.

The algorithm flow designed in this paper includes 4 entities: file uploader (logistics company), cloud server, This keyword is formed by the user encrypting the keyword with a searchable encryption public key, and the function is to query encrypted files in the blockchain database. The detailed steps of the data query algorithm are as follows:

(1) When the user wants to find the encrypted data C_w with the keyword w, he uses a searchable encryption algorithm to generate a trapdoor T_w $Trapdoor\ k_{pr}, w$, and sends a search request to the consensus node on the blockchain database.

(2) After receiving the search request, the consensus node on the blockchain database extracts the trapdoor from the search request, and then executes the searchable encryption formula $b\ Test\ k_{pu}, C_w, T_w$ to match the result. If b 1, the query is successful, and b 0 indicates the query failed.

(3) The user receives the transaction sheet returned from the blockchain database, obtains the encrypted file containing the keyword w from the returned transaction sheet, and then decrypts it with the key to obtain the plaintext data file. If the user wants to verify whether the stored medical data file has been tampered with, he can calculate the hash value of the encrypted file. If the hash value obtained is the same as the hash value recorded in the transaction sheet, the file is correct. The pseudo code of encryption and data query algorithm is as Algorithm 1.

D. SUB-ALGORITHM DESIGN

This paper is based on the hyperledger-fabric framework as an application. Use MSP to effectively manage user IDs and authenticate peers who want to join the network. Hyperledger-fabric relies on MSP components. MSP is based on the Certificate Authority (CA), which generates, verifies and revokes identity-related certificates. Fabric allows the use

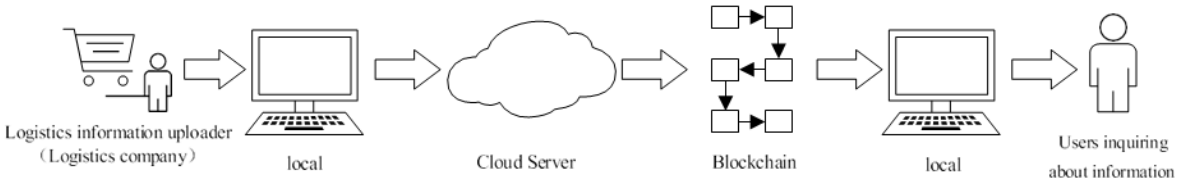

FIGURE 1. Searchable encryption application process.

Algorithm 1 Asymmetric Searchable Encrypted Data Query Algorithm

Input: a security parameter λ.
Output: Document identifiers *id* (D_i).

while Generate random keys K, k_s, k_{pr}, k_{pu}, $\{0,1^{\lambda}\}$, send the public key k_{pu} to the MSP **do**
  if passed the CA verification **then**
    get Signature *(Sig)*
  else
    Exit
  end if
  for $1 \le t \le |w|, 1 \le u \le |D|, a \leftarrow H_K(W_t)$ mod P, Compute $Enc_K(id(D_n))$, store it in $A[1][t]$, get Index table I and encrypted set of documents **do**
    if the client is authenticated using *Sig* **then**
      The Query function of the SC is updated with the I.
    else if the client proposes a transaction $(E_K(D)$ or $T_w)$ **then**
      start PBFT, The latest transaction is validated and append to the blockchain network.
    else if a trapdoor (T_w) transmitted by the client, $d == H_{k_s}(c \cdot a^{-1} mod P)$ **then**
      $X[] \leftarrow Enc_K(id(D_i))$, Encrypted document identifiers $\overline{Enc}_K(id(\overline{D}_i))$.
    else
      get $Dec_K(X[o])$, Document identifiers *id* (D_i)
    end if
  end for
end while

of default interfaces, namely Fabric CA API or external CA. The design process of the two encryption sub-algorithms is as Algorithm 2.

(1) *KeyGen*: Given the security parameter λ, the key root algorithm generates the master key K, asymmetric key pair k_{pr}, k_{pu}, and session key k_s, where K, k_{pr}, $k_{pu} \in (0,1)^{\lambda}$ and k_s are shared with the cloud server. The pseudo code of this process is as Algorithm 2.

(2) *SigGen*: Given the public key k_{pu}, the algorithm generates a unique signature corresponding to the client with the help of MSP and CA. The pseudo code of this process is as Algorithm 3.

Algorithm 2 *KeyGen*

Input: a security parameter λ.
Output: Document identifiers *id* (D_i).

if start *KeyGen* **then**
  Generate random keys K, k_s, k_{pr}, k_{pu} $\{0,1^{\lambda}\}$
end if

Algorithm 3 *SigGen*

Input: a public key k_{pu}.
Output: Signature *(Sig)*.

if start *SigGen* **then**
  Send the public key k_{pu} to the MSP, the CA computes the signature corresponding to the client.
end if

E. DATA INSERTION ALGORITHM DESIGN

After completing the initial setup of the client application and related blockchain network, encrypted documents can be added to the ledger. The client must generate a secure index table I before encrypting documents and storing them on the cloud server. The index Boolean I is a mapping that shows the presence or absence of keywords in the document. This is the only information that can be inferred from I, and the keyword cannot be identified. Once the index table I is successfully generated, the client uses the master key K and the encrypted document D, and then sends the index table I and the encrypted document D to the cloud server to record the encryption process to the ledger. The customer must be verified by MSP. The client sends its signature to the MSP to verify and grant access. The data insertion algorithm is as follows:

(1) Build _Index: The index table I is generated by the client according to the algorithm proposed in [20]. The algorithm uses a cryptographic hash function $H: \{0,1^{\lambda}\} \times W \rightarrow \{0,1\}^{L}$, where L is the length of the output. The keyed hash function H uses the client's master key K to generate a hash of the key. The algorithm is also based on the modular inverse feature, which helps to map probability trap gates. By default, the index table shows the frequency of occurrences of encrypted keywords in documents that caused statistical analysis attacks. In order to avoid this situation, the proposed algorithm hides these values to reduce the risk of attack, and only shows whether there are encrypted

keywords in the document. The pseudo code of this process is as Algorithm 4.

Algorithm 4 *Build _Index*

Input: A set of documents D, dictionary of keywords W and the master key K.

Output: Index table I and encrypted set of documents.

while Initialize dynamic 2D Array A, prime number P of size λ + 1 bits, Build Index I **do**
  for $1 \le t \le |W|$ **do**
    let $a \leftarrow H_K(W_t)$ mod P,
    Compute a^{-1} and store it in $A[1][t]$,
    Compute $Enc_K(id(D_n))$, store it in $A[t][1]$.
    if $1 \le u \le |D|$, $W_t \in D_u$ **then**
      set $A[u][t] = A[u][t] + 1$, $Enc_K(D_u)$.
    else if $1 \le m \le |W|$, $1 \le n \le D$ **then**
      Choose R in Z_p
    else
      $A[n+1][m+1] = A[n+1][m+1] \times R$
    end if
  end for
end while

(2) *Adopt*: Given the index table I and signature, the algorithm embeds the updated index table obtained from the *Build Index* algorithm into the query function of the smart contract. The pseudo code of this process is as Algorithm 5.

Algorithm 5 *Adopt*

Input: The signature (*Sig*), inverted index table I.

Output: The latest inverted index table I is embed within the SC.

if start *Adopt* **then**
  The client is authenticated using *Sig*,
  The Query function of the SC is updated with the I.
end if

(3) *Record* : Given the encrypted document set D, the algorithm appends the documents to the ledger. This requires the client to first use MSP for authentication, which triggers the consensus mechanism. After verifying the transaction node, add it to the ledger. This is necessary for storing encrypted documents, maintaining trap doors, and historical search results. The pseudo code of this process is as Algorithm 6.

F. KEYWORD SEARCH ALGORITHM DESIGN

Keyword search requires the client to generate a probability trapdoor. Since only authorized personnel can generate a meaningful trapdoor, the generation of the trap door requires the client's master key, and the client can also provide keywords for reference search. Probabilistic trapdoors can resist attacks in a differentiated manner, because a new trapdoor can be generated again by the same keyword search, and the generated trapdoor is sent to the cloud server to search on behalf of the client.

Algorithm 6 *Record*

Input: The signature (*Sig*), encrypted documents $Enc_K(D)$.

Output: The latest transaction is validated and append to the blockchain network.

if start *Record* **then**
  The client is authenticated by the MSP through CA using *Sig*, The client proposes a transaction $E_K(D)$ or T_W,
  The endorsing peers validate and endorse the transaction,
  The ordering service orders the transaction into blocks,
  The transaction specific information is broadcast to the peers.
end if

The client application uses the query command to call the cloud server, uses the smart contract of the trapdoor to identify the encrypted document identifier $Enc_K(id(D_i))$ containing the keyword, informs the client of the search result and triggers the Get operation. The trapdoor is attached to the ledger through a consensus mechanism, the required encrypted document is directly retrieved from the blockchain network, and the client using the master key can decrypt the document.

(1) *Generate_Trap*: In order to search for keywords on encrypted documents stored on the blockchain, the client generates a probability trapdoor. Use probabilistic asymmetric encryption algorithm to make the trapdoor probabilistic. The algorithm also uses a keyed hash function similar to the *Build_Index* algorithm. The trapdoor T_w is transmitted to the smart contract to search on behalf of the client. The pseudo code of this process is as Algorithm 7.

Algorithm 7 *Generate_Trap*

Input: The master key K, the session key k_2, keyword w, Hash function $H(.)$.

Output: Transmit T_w to SC.

if start *Generate_Trap* **then**
  let $b \leftarrow Enc_K(w)$ mod P,
  let $a \leftarrow H_K(w)$ mod P,
  let $c \leftarrow a_i b$ mod P,
  let $d \leftarrow H_{k_2}(b)$,
  $T_w \leftarrow (d, c)$.
end if

(2) *Search_Outcome*: The algorithm is embedded in the smart contract and is used to send probabilistic trapdoors and search on behalf of the client. The cloud server calculates and identifies the item d H_{k_2} c a^{-1} mod P. Upon successful identification of the column, the smart contract returns the document identifier to the client of the required block on the network. Since the search is also recorded as a transaction, the Record algorithm is triggered again. The pseudo code of this process is as Algorithm 8.

Algorithm 8 *Search Outcome*

Input: A trapdoor T_w transmitted by the client, a session key k_z, Hash function $H(\ )$ and the index table I.
Output: Encrypted document identifiers $Enc_K(id(D_i))$.

```
while Initialization, generate A dynamic Array X do.
for 1 ≤ l ≤ size of I do
        if d == H_kz  c * a^-1 mod P  then
            X[] ← Enc_K (id (D_i))
        else
            Exit
        end if
    end for
end while
```

(3) *Dec*: The client decrypts the encrypted document identifier to uncover the block containing the desired document. The client can now retrieve the required documents directly from the blockchain network. The pseudo code of this process is as Algorithm 9.

Algorithm 9 *Dec*

Input: Master key K and a set X of encrypted document IDs.
Output: Document identifiers $id(D_i)$.

```
while stsrt Decryption do.
    for 1 ≤ o ≤ size of X do
        Dec_K (X [o])
    end for
end while
```

In this section, based on the logistics scene combined with Blockchain, a logistics chain querying logistics information process using searchable encryption algorithms is designed, and then the corresponding encryption and decryption algorithms are explained. In order to ensure that the logistics information can be updated on the chain in real time, a data insertion algorithm is designed. Finally, according to the characteristics of the searchable encryption algorithm, the process of the keyword search algorithm is given.

I. ALGORITHM ANALYSIS AND PROOF

In this paper, while realizing secure query information, it also satisfies ciphertext security and keyword security.

Theorem 1: Correctness. If the cloud server, user and smart contract execute this algorithm honestly, then the user can get the correct clear text of logistics information.

Proof: Assume that the logistics company has the encryption key k_z, the asymmetric key k_{pr}, k_{pu} to generate the index, the plaintext D of the logistics information data, the keyword set W, and the identification ID of the logistics information data file. Upload the ciphertext ED of logistics data to the cloud server, $ED = DEnc(k_z, D)$. Execute the *Build _Index* algorithm to upload the index I to the cloud server. When n users make a search request, the logistics company randomly selects x_i $(i = 1, 2, \cdots, t-1)$ and calculates $y(x) = k_z + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1} \bmod q$; randomly selects s_i $(i = 1, 2, \cdots, n)$ and the coefficient b_i $(i = 1, 2, \cdots, t-1)$, and then calculates $v(s) = k_1 + b_1s + b_2s^2 + \cdots + b_{t-1}s^{t-1} \bmod q$. Send $y(x_i)$, $v(s_i)$ $(i = 1, 2, \cdots, n)$ to the corresponding user, and store the hash value of $y(x_i)$, $v(s_i)$ and the value of x_i, s_i in the blockchain. When there are users who meet the threshold (set as t) who want to cooperate to search for logistics information containing the keyword w, a two-stage process is required:

(1) Search phase. The user sends his key share $y(x_i)'$ $(i = 1, 2, \cdots, t)$ and keywords w to the blockchain. Since the user and the smart contract execute this scheme honestly, the $k_2{}^r$ calculated by the smart contract according to the Lagrange interpolation polynomial is equal to the key k_2 for generating the index. Then send the calculated search credentials $T^r = (s_w, t_w)$ to the user. The user sends the search credentials T^r to the cloud server, because the cloud server honestly executes this solution, and the cloud server parses $T^r = (s_w, t_w)$. If the index I contains keywords w, the cloud server can find the corresponding EID_w according to s_w, decrypt it to obtain $ID_w = IDDec(t_w, EID_w)$, and finally return the correct ciphertext ED_w of the logistics information containing the keyword to the user according to ID_w.

(2) Decryption stage. When the user wants to decrypt, he sends his key share $v(s_i)'$ $(i = 1, 2, \cdots, t)$ and ciphertext ED_w of logistics information to the blockchain. Since the user and the smart contract execute this algorithm honestly, the $k_1{}^r$ calculated by the smart contract according to the Lagrangian interpolation polynomial is equal to the encryption key k_1, so $D^r{}_w = DDec\ DEnc(k_1, D_w)$, $k_1{}^r = DDec((k_1, D_w), k_1) = D_w$. The smart contract sends D_w to the user, so the user can get the correct plaintext of logistics information data.

Theorem 2: The confidentiality and integrity of logistics information data.

Proof: The logistics information data is encrypted before being uploaded to the cloud server. Although the cloud server is semi-trusted, it will execute the user's request and is also very interested in the user's private data. However, in the solution in this paper, all ciphertext files are stored, and the cloud server cannot obtain the decryption key. Therefore, the file cannot be decrypted, so the privacy of the data can be guaranteed. The client uses the user's public key to encrypt the information. Since k_{pr} is the user's private key, only the user can decrypt it, which realizes the confidentiality of logistics information. In addition, the data in the blockchain is immutable. If the data is already on the chain, it cannot be edited or deleted. The new block constructed has the signature of the block producer, thus realizing the integrity of the logistics information data.

Theorem 3: If the hash function is collision-resistant and the asymmetric searchable encryption scheme is safe, then the scheme proposed in this paper is safe in the sense of adaptive indistinguishability.

Proof: Given the safety parameters λ, A is a polynomialtime adversary, denoted as A $A_0, A_1, \ , A_{q-1}$ $(q \ N)$. The challenger C executes the *KeyGen* algorithm to generatethe key k_{pu}, k_{pr}.

(1) when q 0, the adversary outputs D_0, D_1 and st_{A0} according to the safety parameters λ. The challenger C ran- domly selects b 0, 1 , executes algorithm *DEnc*, encrypts D_b with a key k_{pu} to obtain ED_b, and outputs ED_b and an index Y equal to the real index I_b to the adversary A_1.

Because st_{A0} does not contain k_{pu}, the adversary can-not distinguish ED_b from the real ciphertext, which satis- fies the indistinguishability of ciphertext. Similarly, because st_{A0} does not contain k_{pr} , the adversary cannot distinguish between the index I_b and the real index, which satisfies the indistinguishability of the index.

(2) when $q = 1$, the adversary A_1 outputs $w_{0,i}$, $w_{1,i}$, and st_{A_1} according to st_{A_0} and I_b, ED_b returned by the challenger C. Send $w_{0,1}$, $w_{1,1}$ to the blockchain, the challenger C ran- domly selects $w_{b,1}$, calculates $s_{wb,1}$, $t_{wb,1}$, and outputs the search credentials $T_{b,1}$, $s_{wb,1}$, $t_{wb,1}$ and sends it to the adversary A_2.

Because st_{A_0} does not contain k_{pu}, based on the collision resistance of the hash function, it is computationally infea- sible to generate the same search voucher, so the adversary cannot distinguish $T_{b,1}$ from the real search voucher, which satisfies the indistinguishability of the search voucher.

Through the first theorem, the correctness of the algo- rithm flow is explained; the second theorem analysis shows the completeness of the algorithm in this paper, that is,the algorithm in this paper will not destroy the original data; the third theorem shows from the perspective of adaptive indistinguishability To improve the security of the algo-rithm, the algorithm designed in this paper is effective and feasible.

II. CONCLUSION

In response to the current demand for logistics information at any time, In order to ensure the reliability and privacyof information, combining the advantages and characteris- tics of blockchain technology, and using searchable encryp- tion to encrypt and decrypt data, a logistics information blockchain data query algorithm based on searchable encryp- tion is proposed. The algorithm first encrypts the infor- mation, the encrypted information is stored in the cloud server, and an index list is generated for each group of information, and keywords can be used to query the corre- sponding information. In this paper, the process of encryp- tion and decryption as well as the process of data insertion and data query are designed in detail. Finally, the solu- tion of this paper is analyzed from three aspects of cor- rectness, completeness and security, which proves the fea- sibility of the algorithm of this paper. The next research direction is to conduct in-depth research on the smart con- tract technology in the algorithm to better improve query efficiency.

REFERENCES

[1]. S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Eletronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[2]. Q. F. Shao, ''Blockchain: Architecture and research progress,'' Chin. J. Comput., vol. 425, no. 5, pp. 3–22, 2018.
[3]. Y. Yuan, X. Ni, S. Zeng, and F. Wang, ''Blockchain consensus algorithms: The state of the art and future trends,'' Acta Automat. Sinica, vol. 44, no. 11, pp. 2011–2022, 2008.
[4]. D. X. SONG, D. WAGNER, and A. PERRIG, ''Practical techniques for searches on encrypted data,'' IEEE Symp. Secur. Privacy., Oct. 2000, pp. 44–55.
[5]. J. Li, C. F. Jia, Z. Liu, J. Li, and M. Li, ''Survey on the searchable encryption,'' J. Softw., vol. 26, no. 1, pp. 109–128, 2015.
[6]. D. Boneh, ''Public key encryption with keyword search,'' in Proc. Int. Conf. Adv. Cryptol., 2004, pp. 506–522.
[7]. Q. J. Zheng, S. H. 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.
[8]. J. Shao, Z. Cao, X. Liang, and H. Lin, ''Proxy re-encryption with keyword search,'' Inf. Sci., vol. 180, no. 13, pp. 2576–2587, Jul. 2010.
[9]. M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, ''Comprehensive study of symmetric key and asymmetric key encryption algorithms,'' in Proc. Int. Conf. Eng. Technol. (ICET), Aug. 2017, pp. 1–7.
[10]. X. Liu, X. Yu, X. Ma, and H. Kuang, ''A method to improve the freshdata Query efficiency of blockchain,'' in *Proc. 12th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Feb. 2020, pp. 823–827.
[11]. Y. P. Luo, N. T. Zhu, C. W. Mao, and J. X. Ch, ''A practical searchable encryption scheme based on connection keywords,'' *Comput. Eng.*, vol. 46,no. 2, pp. 175–182, 2020.
[12]. N. Zh and Z. Sh, ''Mechanism of personal privacy protection based on blockchain,'' *Comput. Appl.*, vol. 37, no. 10, pp. 2787–2793, 2017.
[13]. L. G. CH and Q. LI, ''Blockchain data privacy protection mechanism basedon searchable encryption,'' *Comput. Appl.*, vol. 39, no. 2, pp. 140–146, 2019.
[14]. K. Zh and G. Zh, ''A study of ciphertext full-text retrieval based on search-able encryption in cloud environment,'' *Comput. Appl. Softw.*, vol. 30,no. 4, pp. 35–41, 2017.
[15]. H. Qin, Z. Li, P. Hu, Y. Zhang, and Y. Dai, ''Research on point-to-point encryption method of power system communication data based on block chain technology,'' in *Proc. 12th Int. Conf. Intell. Comput. Technol. Autom.(ICICTA)*, Oct. 2019, pp. 328–332.
[16]. S. F. Niu, W. K. Liu, and L. X. Cheng, ''Electronic medical record data sharing scheme based on searchable encryption via consortium blockchain,'' *J. Commun.*, vol. 41, no. 8, pp. 204–214, 2020.

[17]. L. Zhang, Z. Y. Zheng, and Y. Yuan, ''A controllable sharing model for electronic health records based on blockchain,'' *J. Automat.*, vol. 4,pp. 1–14, Nov. 2020.
[18]. J. Sun, L. Ren, S. Wang, and X. Yao, ''Multi-keyword searchable anddata verifiable attribute-based encryption scheme for cloud storage,'' *IEEEAccess*, vol. 7, pp. 66655–66667, 2019.
[19]. S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan, and C. Glackin, ''A new secure and lightweight searchable encryption scheme over encrypted cloud data,'' *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 4,pp. 530–544, Oct. 2019.