Multi-keyword Ciphertext Sorting Search Based on Conformation Graph Convolution Model and Transformer Network in English Education

Hang Li[1], Zeyang Li[1], Xiaowei Wang[1], Muhammad Ibrar[1], and Xinjie Zhu[2]
(Corresponding author: Zeyang Li and Xiaowei Wang)
Software College, Shenyang Normal University[1]
Shenyang 110034, China
Email: lihangsoft@163.com
School of Foreign Languages, Zhengzhou University of Science and Technology[2]
Zhengzhou 450000, China
(Received Nov. 26, 2023; Revised and Accepted Feb. 7, 2024; First Online June 22, 2024)
The Special Issue on Data Fusion, Deep Learning and Optimization Algorithms for Data Privacy Protection
Special Editor: Prof. Shoulin Yin (Harbin Institute of Technology)

Abstract

With the wide application of cloud computing, the outsourcing service model for data or computing is more and more accepted by the industry. Because asymmetric searchable encryption is difficult to deal with the problem of ciphertext sorting, the existing research on multi-keyword ciphertext sorting search mainly adopts symmetric searchable encryption mechanism, the core problem of which is the data structure, construction algorithm, and search algorithm of the secure, searchable index. Therefore, a new multi-keyword ciphertext sorting search based on the conformation graph convolution model and Transformer network is constructed in this paper. First, the Transformer network uses the bottleneck features of input samples to generate the bottleneck features of pseudo-abnormal data, thereby adding abnormal data information to the training set. Then, the model constructs a multi-modal feature learning and fusion graph convolutional network to obtain contextual features of each piece of information. The security and performance analysis show that the proposed scheme is safe and feasible under the known ciphertext model. Simulation results show that the proposed scheme can realize result verification and fair payment at an acceptable cost.

Keywords: Conformation Graph Convolution; Feature Learning; Multi-Keyword Ciphertext Sorting Search; Transformer Network

1 Introduction

With the rapid development of cloud computing, more and more users migrate large amounts of data to the cloud platform to save local storage costs. At the same time, the cloud platform also provides instant services for remote storage and computing, which is convenient for users to access and use data anytime and anywhere. However, since users have lost control of the data, how to ensure data privacy security has become a key issue. In order to protect the privacy of sensitive data, the data needs to be encrypted before users upload it to the Cloud Service Provider (CSP) [8]. However, data encryption makes plaintext based keyword retrieval technology impossible to use. The proposed searchable encryption technology can not only realize the retrieval of encrypted data, but also ensure the privacy and security of data.

In searchable encryption, it is often assumed that CSP is honest and curious entities. In practice, however, in order to save computing resources while obtaining service fees, CSP may dishonestly perform search operations and send incorrect or incomplete search results to users [17, 30]. In the pay-before-use model, even if the above dishonesty occurs, the user must first pay the service fee to the CSP. In the use-before-pay-later model, there may be cases where dishonest users receive the correct results and refuse to pay the service fee. In the pay-before-use model, the value of the Data provided by the data owner to the CSP needs to be taken into account in the transaction, that is, the data user should pay the message fee for the data provided to the data owner when conducting a transaction [5]. To address these equity issues, traditional solutions often rely on a trusted third party. However, because the third party does not have the ability to directly verify the correctness and integrity of the search results, when there is a transaction dispute, the third party needs to spend a lot of time to solve the dispute, the fair payment cannot be directly guaranteed,

and the dishonest behavior generated in the transaction process will not be punished. On the other hand, when third-party organizations, CSP and users interact with each other, users' personal privacy information on third-party organizations is also at risk of being leaked.

From the above analysis, we can see that an effective method is needed to solve the fair payment problem in traditional searchable encryption schemes. With the advent of Bitcoin [20], blockchain as its underlying technology can provide support for solving this problem. Blockchain has the characteristics of decentralization and immutability, which can be well combined with cloud computing; Smart contracts on blockchains [9, 24] can be written directly into code and executed automatically, outside the control of any centralized authority. Thus, blockchains and smart contracts are suitable for performing verification operations and enabling fair payments in searchable cryptographic schemes.

In the process of combining blockchain smart contracts with searchable encryption schemes, people use blockchain smart contracts to perform verification of search results and achieve fair payment. In addition, some schemes use smart contracts to replace CSP to perform search operations. In the execution process, blockchain smart contracts need to carry out multiple transaction transactions, store indexes that take up more space and perform complex search operations, which has low scalability, high cost and large time consumption. And the cost of fees and time increases as the complexity of the operations performed by smart contracts increases. Therefore, it is necessary to consider reducing the complexity of the operations performed by smart contracts on the basis of ensuring the realization of result verification and fair payment to reduce the overhead of time and expense costs, improve efficiency, and expand the function of the scheme to be more user-friendly [16, 18, 31].

In addition, in practical applications, the schemes that only support single keyword retrieval often can not meet the needs of users. For example, in order to obtain more accurate search results, users typically enter multiple keywords when searching and want to return the first k documents that are most relevant to the entered keywords [2,32]. Therefore, it is necessary to consider designing a searchable encryption scheme with richer retrieval functions on the basis of combining blockchain technology to ensure the realization of result verification and fair payment.

In order to reduce the time and cost of realizing result verification and fair payment, our work goal in this paper is that this scheme combines the powerful and efficient retrieval ability of CSP with the advantages of blockchain smart contract to automatically execute contract contents, and realizes the sequential retrieval of ciphertext, the verification of search results, and the fair payment among data owners, CSP and data users.

2 Related Works

Searchable encryption technology is a kind of password primitive that allows users to search ciphertext data. It uses the powerful computing resources of cloud server for keyword search, and its core idea is that users have the ability to search for keywords in ciphertext domain. Mihailescu *et al.* [14] proposed the idea of searchable encryption for the first time to solve the problem of searching encrypted data on the cloud platform. Subsequently, searchable encryption under cloud storage technology became a research hotspot. In recent years, many efforts have been made to enrich the functions of searchable encryption, and schemes such as multi-keyword search [22], dynamic encryption search [27], fuzzy keyword search [4] and verifiable encryption search [11] have been proposed successively. These schemes usually assume that the cloud server will honestly perform the task, however, the cloud server is often not completely trustworthy, it may save computing resources or defrauds the service fee, after receiving the service fee to return incorrect or incomplete results; at the same time, even if the user receives the correct search results, if the user claims that the search results are incorrect, it may maliciously refuse to pay the service fee. The above situation leads to service-payment inequity, resulting in distrust between users and cloud servers. In order to solve the above problems, traditional solutions usually consider supervision and arbitration by a trusted third party.

Since its inception, blockchain has received a lot of attention from academia and industry for its ability to enable fair payments without the introduction of third-party institutions. Therefore, there are active attempts to combine blockchain with searchable encryption technology to solve the problems of traditional solutions. Niu *et al.* [15] proposed a trustworthy keyword search scheme based on cloud storage, which used bitcoin blockchain technology and hash function to achieve fair payment of search costs without a third party. The scheme established a secure index based on digital signature, which guaranteed the correctness of the retrieval results at the client side and verified the validity of the encrypted data at the server side. However, in the process of verifying the correctness of the result, the scheme needed a lot of signature verification calculation, and the user cost was high. Gao *et al.* [7] designed a fair symmetric searchable encryption scheme based on the Bitcoin blockchain, which automatically verified the search results through the blockchain to ensure the fairness of transactions between users and cloud servers. However, the scheme needed to execute six transactions each time to obtain the search results, and the verification of the search results was realized through the Bitcoin script. The transaction cycle was too long, resulting in high time cost. All the schemes in references [26,29] were searched by cloud servers, and the search results were verified based on the Bitcoin blockchain to achieve fair payment between multiple parties. However, because Bitcoin smart contracts were not Turing-complete, their

functions were limited, the transaction process was complex, the transaction cycle was long, and the efficiency was not high. Ali *et al.* [1] implemented dynamic and efficient keyword search in a distributed storage network, used smart contracts to record encrypted search logs on the Ethereum blockchain, and designed a protocol to handle disputes and issue commissions for fair search between clients and servers. The scheme was a retrieval operation performed by contracted service nodes in a distributed storage network, with searchable indexes and metadata of search results anchored to the blockchain as evidence. Arbitration nodes on the arbitrator shard in the distributed storage network checked the correctness of the search results and realized fair payment based on the Ethereum smart contract. When a data user applies for arbitration, each arbitration node needed to re-execute the search algorithm independently and determine whether the judgment request issued by the client (i.e. the stop payment request) was valid based on the re-generated search results. These individual arbitration results were then pooled into the arbitration smart contract to make a final decision. If more than 2/3 of the nodes in the arbitrator shard accepted the judgment, the time-limited payment was suspended. As we can see, this arbitration process would waste a lot of computing resources.

Li *et al.* [13] proposed a blockchain-based searchable encryption scheme that supported complex logical expression queries, which used Ethereum smart contracts to ensure the correctness of the search results and could achieve fair payment without any verification mechanism. Guo *et al.* [11] proposed a blockchain-based distributed storage system that supported fine-grained access control. The system used Ethereum smart contracts to realize keyword search function in ciphertext state, which solved the problem that the cloud server in the traditional cloud storage system should not be able to return all search results or return wrong results. Su *et al.* [21] designed an attribute-based search encryption scheme based on blockchain and supporting verification. The scheme designed search contracts and verification contracts based on Ethereum smart contracts, and realized fair payment supporting multi-keyword search without requiring additional local verification. Wang *et al.* [23] and Bi *et al.* [3] were all encrypted index and search results stored by blockchain, and search operations were performed and fair payment was achieved based on Ethereum smart contracts. However, the storage capacity of the blockchain was limited by the nodes with the smallest storage space, and complex encrypted indexes needed to be fragmented before they could be stored into blockchain transactions, and these transactions could be uploaded one by one, which would consume a lot of time. In addition, the smart contract would consume a certain amount of costs when performing operations. In the scheme, the search operation and verification operation of high complexity were performed by the smart contract, which required a large amount of calculation when executing, and would also lead to increased costs. Therefore, the above schemes have the problems of low scalability, high time cost and high cost. In addition, they only support single keyword search, and do not consider the correlation ranking of search results, which is not flexible in function and not user-friendly.

Therefore, this paper proposes a new multi-keyword ciphertext sorting search based on conformation graph convolution model and Transformer network. Using the cloud server's efficient retrieval ability and the automatic execution of Ethereum smart contracts, the cloud server stores the encrypted index tree and lookup table; The simultaneous execution of the search algorithm can effectively reduce the complexity of the smart contract execution operation, thus reducing the time cost and expense costs consumed; The verification process of the results achieved by the Ethereum smart contract not only ensures the correctness and integrity of the search results, but also completes the fair payment between the data owner, the cloud server and the data user, and can effectively reduce the cost and improve the verification efficiency. In addition, this paper uses balanced binary tree as the index, and realizes the dynamic update of multi-keyword search and the ranking of search results on the basis of ensuring the efficiency of search, which improves the flexibility and user friendliness of the scheme.

3 Data Training Based on Transformer Network

Transformer network is composed of feed-forward neural network. Its purpose is to find another feature space Z_t that is far away from the feature space Z corresponding to the input sample, and transform the bottleneck feature $z \in Z$ of the input sample into pseudo-anomaly bottleneck feature $z_t \in Z_t$. The Transformer network is defined as $f_T(\cdot) : Z \to Z_t$, then:

$$z_t = f_T(z). \tag{1}$$

From formula (1), we can get the pseudo-abnormal bottleneck feature z_t in the feature space which is far away from the bottleneck feature z of normal data. Therefore, z_t is regarded as a bottleneck feature with abnormal data. The model adds abnormal data to the training set by getting z_t.

By minimizing the reconstruction error of sample x, the model enables encoder E_1 to obtain better bottleneck characteristics, and thus obtains better reconstructed samples by decoder D_1, which is expressed as follows:

$$\min_{\theta_E, \theta_D} ||x - \hat{x}||_2^2. \tag{2}$$

Where $\hat{x}$ is the reconstructed sample. θ_E, θ_D are the parameter sets of encoder and decoder.

In order to make The Transformer network obtains a feature space Z_t that is far away from the normal data feature space Z, and generates a pseudo-abnormal bottleneck feature with abnormal data information. The

Transformer network is trained by maximizing the error between the bottleneck feature z of the input sample and the bottleneck feature z_t transformed by the Transformer network. It is expressed as follows:

$$\max_{\theta_T} ||z - z_t||_2^2. \tag{3}$$

Where θ_T is the parameter set of Transformer network.

Furthermore, in order to enable the decoder to map the bottleneck feature with abnormal data information to normal data rather than itself as much as possible, the model causes the decoder D_2 to map the transformed bottleneck feature z_t to $\hat{x}_t$ by minimizing the error between $\hat{x}$ and $\hat{x}_t$, making b$\hat{x}_t$ as similar as possible to the normal data. It is expressed as follows:

$$\min_{\theta_E,\theta_D} ||\hat{x} - \hat{x}_t||_2^2. \tag{4}$$

In order to further improve the reconstruction error of abnormal data, the model enables encoder E_3 to obtain bottleneck feature $\hat{z}_t$ from $\hat{x}_t$ by minimizing the error between z_t and $\hat{z}_t$, so that $\hat{z}_t$ is as similar as possible to bottleneck feature z_t of normal data, which is expressed as follows:

$$\min_{\theta_E,\theta_D} ||\hat{z} - \hat{z}_t||_2^2. \tag{5}$$

Comprehensively considering equations (2)-(5), the model training objective of the method in this paper is to minimize the loss function:

$$\begin{aligned}\min_{\theta_E,\theta_D,\theta_T} \frac{1}{N}(&||x - \hat{x}||_2^2 \\ &+ \alpha||x - \hat{x}||_2^2 \\ &+ \beta||\hat{z} - \hat{z}_t||_2^2 \\ &+ \gamma||z - z_t||_2^2)\end{aligned} \tag{6}$$

Where N is the number of training samples. α, β, γ are the weights of each loss function. Encoders E_1, E_2, and E_3 use the same network structure and share parameters. Decoder D_1 and D_2 use the same network structure and share parameters.

Assuming that the given test sample is normal data, the encoder maps it to the bottleneck feature of the normal data, and then the decoder maps it back to the normal data, giving the normal data a small reconstruction error. Assuming that the given test sample is abnormal data, the encoder maps it to the bottleneck feature of the abnormal data, and the decoder decodes the bottleneck feature with abnormal data information into normal data as much as possible instead of reconstructing itself, so that the abnormal data can obtain a large reconstruction error. Therefore, the transformer method can use the reconstruction error of samples as the anomaly score to classify samples.

Since the encoder used in the training phase has the same structure and shared parameters, the decoder also has the same structure and shared parameters. Therefore, the test phase only needs to use any set of trained encoders and decoders to form a new model and classify the test samples. Given a test sample x_{test}, the reconstruction sample $\hat{x}_{test}$ of x_{test} is obtained by the new model, the reconstruction error of x_{test} is calculated, and the reconstruction error is used as the anomaly score $S(x_{test})$ to classify x_{test}, which is expressed as follows:

$$S(x_{test}) = ||x_{test} - \hat{x}_{test}||_2^2. \tag{7}$$

The training process based on Transformer network is shown in **Algorithm 1**.

Algorithm 1 Transformer training

1: **Input:** training set $X = x_i{}_{i=1}^{N}$.
2: Output: encoder $f_E(\cdot)$ and decoder $f_D(\cdot)$.
3: Initialize parameter sets θ_E, θ_D, θ_T of encoder $f_E(\cdot)$, decoder $f_D(\cdot)$ and Transformer network $f_T(\cdot)$.
4: for $i = 1$ to N do
5: Through equation (2), the training sample x_i is calculated and the bottleneck feature z is obtained after encoding.
6: Equation (3) is used to calculate the bottleneck feature z_t of z after Transformer network transformation.
7: The decoded samples $\hat{x}$ and $\hat{x}_t$ of z and z_t are obtained by equation (3).
8: The bottleneck features $\hat{z}$ and $\hat{z}_t$ of $\hat{x}$ and $\hat{x}_t$ after re-encoding are calculated by equation (2).
9: The bottleneck features $\hat{z}$ and $\hat{z}_t$ of $\hat{x}$ and $\hat{x}_t$ after re-encoding are calculated by equation (3).
10: The parameter sets θ_E, θ_D, θ_T are updated by equation (7) and stochastic gradient descent.
11: End

3.1 Graph-based Feature Learning

Each discourse in the data set is taken as a graph node, and the graph $G = (v, \varepsilon)$ is constructed, where $v(|v| = N)$ represents the discourse node. $\varepsilon \subset v \times v$ is an edge between nodes.

Two nodes can be connected by different edges, representing multiple relationships of the three modal features. In this paper, the weights of nodes u_i and edges between u_j are calculated according to the following circumstances.

- Consider the feature transfer of the same mode between two nodes. Since the same modal features of two nodes are in the same semantic space, the feature transfer can be carried out regardless of whether the nodes come from the same conversation. Weight reuse Angle similarity measurement of edges between two nodes.

$$a_{ij} = 1 - \frac{arccos(sim(x_i^{mod(0)}, x_j^{mod(0)}))}{\pi}. \tag{8}$$

Where $sim(\cdot)$ is the cosine similarity function. $x_i^{mod(0)}$, $x_j^{mod(0)}$ respectively represent the initial features of some same mode of the i and j discourse, $mod \subseteq a, t, v$.

- Considering the feature transfer of different modes between two nodes, two cases can be divided according to whether the two nodes come from a conversation:

  (a) If two nodes come from different conversations, the different modal features are not passed, in which case the weight below is 0. This is because although linear transformations are carried out in the initial feature extraction process of the three modes, the features of the different modes can be considered basically aligned in the semantic space. However, different dialogue scenes and dialogue content are very different, which enlarges the gap between different modes, so this paper thinks that the feature transfer should not be carried out in this case.

  (b) If two nodes come from the same dialogue, the different modal features are also relevant due to the consistent topic and content of the dialogue, and feature transfer is required. The weight of the edge between two nodes is also measured by angular similarity:

$$a_{ij} = 1 - \frac{arccos(sim(x_i^{mod'(0)}, x_j^{mod''(0)}))}{\pi}. \tag{9}$$

  Where $x_i^{mod'(0)}$, $x_j^{mod''(0)}$ represent the initial features of different modes of discourse i and j respectively, $mod', mod'' \subseteq a, t, v$, $mod' \neq mod''$.

  The adjacency matrix is constructed according to the weight calculation method of the edges between the nodes. For a certain modal feature of a node, three kinds of adjacency matrices can be constructed to transfer and learn the feature.

  Taking the feature learning of speech mode a of a node as an example, considering the relationship between speech mode a and its own speech mode a, text mode t and image mode v, three kinds of graph adjacency matrices can be constructed, and the feature matrix $X^{a(0)}$ can be updated.

  In addition, for the feature learning of the text mode t of nodes, three kinds of graph adjacency matrices A^{tt}, A^{ta} and A^{tv} are constructed. For the feature learning of image mode v of nodes, three kinds of graph adjacency matrices A^{vv}, A^{va} and A^{vt} are constructed.

This paper takes the feature learning of node speech mode a as an example to illustrate the feature learning process of different modes. The three graph adjacency matrices A^{aa}, A^{at} and A^{av} are convolution with the initial data feature $X^{a(0)}$ of the node by multi-layer GCN, and the updated three data features $A^{aa(l)}$, $A^{at(l)}$ and $A^{av(l)}$ are obtained by using four-layer GCN for encoding. The specific process is as follows.

For the graph $G = (v, \varepsilon)$, the Laplacian matrix formula for renormalization is as follows:

$$L = \tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}. \tag{10}$$

$$L = (D + I)^{-0.5}((A^{mod'mod''}) + I)(D + I)^{-0.5}. \tag{11}$$

Where D represents the degree matrix. I stands for identity matrix. L stands for the renormalized Tullapras matrix of G. $mod' = a$, $mod" \subseteq a, t, v$. The iterative representation of graph convolutional networks with different layers is:

$$\begin{aligned} X^{mod(l+1)} =& \sigma(((1-\alpha)LX^{mod(l)} + \alpha X^{mod(0)} \\ & (1-\beta^l)I + \beta^{(l)}W^{(l)})). \end{aligned} \tag{12}$$

Where $X^{mod(0)} \in R^{N \times d_0}$ is the initial feature of a certain mode of the graph node, which is $X^{a(0)}$ for multi-mode feature learning. σ is the activation function. $W^{(l) \in R^{d_{l-1} \times d_l}}$ is a learnable weight matrix. In order to solve the problem of excessive smoothing and gradient disappearance caused by the introduction of multilayer graph convolution, the initial feature $X^{mod(0)}$ is added to the high level as residual, and I is added to the weight matrix $W^{(l)}$. α and $\beta^{(l)}$ are two hyperparameters. This article sets $\beta^{(l)} = \log(\eta/l + 1)$, where η is also a hyperparameter. Using DeepGCN with l layers, it can get $X^{mod(l+1)}$.

For feature learning of data modes, $X^{mod(0)} = X^{a(0)}$, there are three kinds of adjacency matrices for feature learning, then $X^{mod(l+1)}$ corresponds to three kinds of features $X^{aa(l+1)}$, $X^{at(l+1)}$, $X^{av(l+1)}$ obtained from the feature learning of three kinds of adjacency matrices. The three features are spliced together to obtain the data modal feature $X^{a(l+1)}$ after feature learning.

$$X^{a(l+1)} = X^{aa(l+1)} \oplus X^{at(l+1)} \oplus X^{av(l+1)}. \tag{13}$$

Here $\oplus$ is concatenation operation.

The vector in row i of the above eigenmatrix is the data feature $x_i^{a(l+1)}$ corresponding to the convolution of a node u_i graph. For the feature learning of the text modes and image modes of nodes, the above multi-level DeepGCN encoding is also used to obtain three convolution features of the text modes. After the features are splicing, $X^{t(l+1)}$ is obtained. Three convolution features of the image modes are obtained, and $X^{t(l+1)}$ is obtained after the features are spliced.

$$X^{v(l+1)} = X^{va(l+1)} \oplus X^{vv(l+1)} \oplus X^{vt(l+1)}. \tag{14}$$

$$X^{t(l+1)} = X^{ta(l+1)} \oplus X^{tv(l+1)} \oplus X^{tt(l+1)}. \tag{15}$$

Finally, the three modal features are combined to obtain the total feature matrix X^{l+1} after convolutional learning.

$$X^{(l+1)} = X^{a(l+1)} \oplus X^{t(l+1)} \oplus X^{v(l+1)}. \tag{16}$$

4 Performance Analysis

The scheme in this paper uses Python to implement the searchable encryption algorithm, in which the pseudo-random function is simulated by HMAC-MD5, and the MAC function is simulated by HMAC-SHA256. Ethereum's simulation experiments are conducted on the Ethereum VirtualMachine (EVM), where an Ethereum smart contract is built using the Solidity language. The computer hardware is configured as Inter Core i5 7300HQ2. 50GHz processor, 16GB RAM, 512GB SSD, and Ubuntu18.04LTS operating system [19].

4.1 Performance Analysis of Key Algorithms

The computational overhead of GenIndex algorithm in DO index generation phase includes the construction of encrypted index tree and lookup table. The computational overhead of the Search algorithm in the CSP retrieval stage includes the retrieval of the encrypted index tree and the positioning of the lookup table. The computational overhead of GenIndex and Search is simulated below. The data set used in the experiment contains 9810 documents in 20 categories, and 5000 documents are selected as the test data in this paper.

In order to observe the relationship between the computation cost of GenIndex performed by DO and Search performed by CSP and the number of documents, the number of documents was set to 500-5000(step size 500) in experiment 1. The experimental results are shown in Figure 1. As can be seen from Figure 1(a), the GenIndex time basically increases linearly as the number of documents increases. In particular, when the number of test documents is 500 and 5000, the GenIndex time is 221.656s and 2012.545s, respectively. It should be noted that this operation can be performed in the offline state, and the time cost is acceptable.

As shown in Figure 1(b), it can be seen that the Search time basically increases logarithmically with the increase in the number of documents. Under the current number of experimental documents, the Search time is all less than 1s, and the time cost is acceptable. In particular, when the number of test documents is 500 and 5000, the Search time is 7.6ms and 50.5ms, respectively.

4.2 Computation Cost Comparison

The calculation cost of index, search and verification is compared between the proposed scheme and relevant comparison schemes, and the results are shown in Table 2. Where E_0 and E_1 represent exponential operations on two different multiplicative cyclic groups, respectively. M_M stands for modular multiplication. H stands for hash operation. F stands for pseudo-random function operation. V represents MAC function operation. L represents the operation that builds each internal node. X represents the dot product between n-dimensional vectors. E and D indicate the encryption and decryption operations, respectively. M stands for modulo operation. SIG indicates the signature algorithm, which includes two processes: signature and verification. $\times$ indicates that it does not participate in this operation. g indicates the number of records of the conditional expression. I_A represents the subscript of the gate access policy. Y indicates the number of leaf nodes in the tree access policy. j indicates the number of returned ciphertext files. a represents the number of copies into which the document identifier that satisfies the conditional expression is divided. b represents the number of predicted steps by the data user. t is the number of keywords in W. n is the size of the keyword dictionary DW. Z is the number of records in the lookup table. θ is the number of documents containing the query keyword. In the real world, θ is much smaller than the document set base m.

In the stage of index generation, the scheme of reference [28] requires one pseudo-random function operation, one hash operation and one signature algorithm operation on the keyword dictionary and the document collection containing the keywords. The reference [12] requires three pseudo-random function operations, one encryption operation, and one hashing operation on the keyword dictionary and the document collection containing the keywords. The scheme of reference [33] requires one hashing operation on the ID of the document and the set of added tags, one encryption operation on the plaintext document set, and one pseudo-random function operation on the keyword dictionary. The data owner of the scheme in reference [25] needs to perform a time-consuming modular multiplication operations and $2(a+1)g$ pseudo-random function operations. Data owners of the references [6, 10] all need to carry out time-consuming exponential and modular multiplication operations. Meanwhile, the reference [6] scheme needs to carry out two pseudo-random function operations on each keyword, and the reference [10] scheme needs to carry out one pseudo-random function operation. In this paper, the balanced binary tree is used as the index, and the retrieval efficiency is high. The construction of internal nodes requires $m-1$ times, the encryption of each internal node requires one symmetric encryption operation for each bit of its vector, and each leaf node vector is encrypted by the secure K-nearest neighbor algorithm, and the multiplication of $n \times n$ matrix and n-dimensional vector needs to be performed twice. Constructing a lookup table that matches the encrypted index tree requires Z pseudo-random function operations and Z MAC function operations.

In the search stage, reference [33] conducts $O(Z)$ times of comparison between the published list and the abstract index, and then conducts m times of hash operation search. The scheme in reference [25] requires 6 time-consuming modular multiplication operations and 26 pseudo-random function operations. The references [6,10] generate a lookup table of key-value pairs as an index, with a search efficiency of $O(Z)$. References [10, 33] all

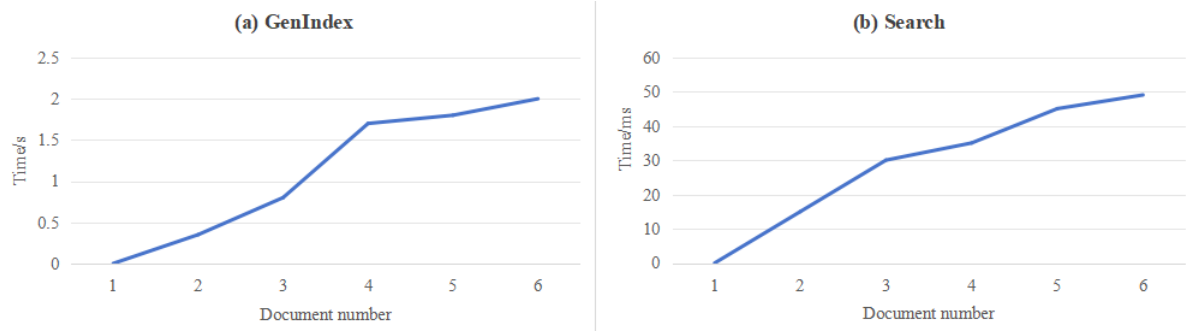

Figure 1: The relationship between the number of GenIndex and Search documents and the computational overhead

Table 1: Computing cost comparative analysis of searchable encryption schemes

Schemes	Index generation phase	Search phase	Verification stage
[28]	$n\theta(F+H+SIG)$	$m(H+SIG)$	$m(H+SIG)$
[12]	$n\theta(3F+H+E)$	$2D+4H+O(Z)$	$4H$
[33]	$2mH+mE+nF$	$mH+2O(Z)$	$(j+m)H$
[25]	$2(a+1)gF+aM_M$	$2bF+bM_M$	$\times$
[6]	$2E_0+E_1+2nF$	$O(Z)$	$\times$
[10]	$M_M+(2Y+1)E_0+E_1+nF$	$O(Z)$	$j!H$
Proposed	$(m-1)L+2nmX+n(m-1)E$ $+ZF+ZV$	$t\theta(lbm-1)D+2\theta X$ $+\theta M+O(Z)$	V

store encrypted indexes and perform search algorithms on the blockchain, resulting in high blockchain overhead, both occupying storage space and low efficiency. However, both the scheme of reference [12,28] and the scheme of this paper use CSP to search, and CSP stores ciphertext documents and encrypted indexes, which can reduce the blockchain overhead and improve the retrieval efficiency. In this scheme, the internal node is decrypted for $t\theta(lbm-1)$ times. In the worst case, the dot product operation between two n-dimensional vectors needs to be performed for 20 times and the modulo operation for θ times. The corresponding *proof* is located in the lookup table according to the *token* of DU, and $O(Z)$ comparison operations are required.

In the stage of verifying the correctness of the results, reference [28] removes the audit institution, but requires users to perform m hashing operations and m signature algorithm operations locally. As the number of documents increases, the verification time increases. The scheme in reference [12] only needs to carry out 4 hashing operations, but the process needs to involve 6 transactions on the Bitcoin script, so the verification time is long and the efficiency is low. The scheme of reference [33] requires each arbitration node on the arbitrator fragment in the distributed storage platform to perform j hashing operations on the set of added tags obtained by the client, and then re-perform the search operation to perform m hashing operations to verify the search results. The proposal in reference [10] and the proposal in this paper use smart contracts to verify search results, and the proposal in reference [10] requires $j!$ For secondary hashing operation, the scheme in this paper only needs to perform one MAC function operation to determine whether the value after MAC function operation is correct and complete the verification.

In summary, compared with other schemes with verification function, the proposed scheme has the lowest computational cost in the verification phase. This is because this scheme uses CSP to store ciphertext documents, encrypt indexes, and perform search algorithms, while smart contracts only need to perform verification and fair payment operations, which makes its computation cost low. However, this also leads to the increase of CSP storage overhead and computing overhead. However, from the analysis of computing overhead in Search phase and the experimental results of search algorithm, it can be seen that the computing overhead in search phase of CSP is acceptable.

In order to explore the relationship between the number of different documents and the retrieval time, we conducted an experiment. The results are shown in Table **??**. It can be seen that the retrieval time of reference [12] basically increases linearly with the increase of the number of documents. The retrieval time of reference [25] is slightly lower than that of reference [33]. However, the retrieval time of this proposed scheme is the fastest due to other schemes. In this scheme, the retrieval time is almost constant when the number of documents increases.

Table 2: Retrieval time comparison of different schemes/s

Scheme	1000	2000	3000	4000	5000	6000
Reference [28]	0.19	0.38	0.57	0.78	1.12	1.28
Reference [12]	0.15	0.22	0.26	0.35	0.42	0.48
Reference [33]	0.12	0.18	0.22	0.29	0.35	0.39
Reference [25]	0.11	0.16	0.21	0.26	0.31	0.36
Reference [6]	0.10	0.10	0.10	0.10	0.10	0.10
Reference [10]	0.03	0.03	0.03	0.03	0.03	0.03
Proposed	0.02	0.02	0.02	0.02	0.02	0.02

5 Conclusion

This paper proposes a new multi-keyword ciphertext sorting search based on conformation graph convolution model and Transformer network, which supports verification and fair payment, and realizes the verification of search results, the fair payment between three parties and the multi-keyword sorting retrieval of ciphertext. In order to realize the verifiability and fair payment of the search results, and reduce the time and cost, the scheme is designed by the cloud server to store the encrypted index tree and lookup table, and perform the search operation. The verification and fair payment of the search results are completed by the Ethereum smart contract, which effectively reduces the complexity of the smart contract execution operation, reduces the time and expense, and improves the verification efficiency. In addition, the scheme uses balanced binary tree as the index, which ensures the retrieval efficiency and realizes the functions of multi-keyword retrieval, ranking of search results and dynamic update, which improves the flexibility and user friendliness of the scheme. Finally, the safety and performance of the scheme are analyzed, and the simulation experiment is carried out. Performance analysis and experimental results show that the proposed scheme is feasible and practical. The results of functional comparison show that compared with the existing blockchain-based searchable encryption schemes, the proposed scheme is more comprehensive in terms of functions. In addition, the verification process of the scheme in this paper is carried out for all the search results, and there is room for further optimization. Future research on verification strategies for specific transactions to better meet user needs while reducing time and expense costs.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Ali, M. Pasha, J. Ali, O. Fang, M. Masud, A. Jurcut, "Deep learning based homomorphic secure search-able encryption for keyword search in blockchain healthcare system: A novel approach to cryptography," *Sensors*, vol. 22, no. 2, pp. 528, 2022.

[2] M. Ali, MR. Sadeghi, X. Liu, Y. Miao, "Verifiable online/offline multi-keyword search for cloud-assisted industrial internet of things," *Journal of Information Security and Applications*, vol. 65, 2022.

[3] J. Bi, S. Yin, H. Li, L. Teng, C. Zhao, "Research on medical image encryption method based on improved Krill Herb algorithm and chaotic systems," *International Journal of Network Security*, vol. 22, no. 3, pp. 486-491, 2020.

[4] B. Deebak, F. Memon, K. Dev, S. Khowaja, *et al.*, "AI-enabled privacy-preservation phrase with multi-keyword ranked searching for sustainable edge-cloud networks in the era of industrial IoT," *Ad Hoc Networks*, vol. 125, 2022.

[5] C. K. Dehury and P. K. Sahoo, "Failure aware semi-centralized virtual network embedding in cloud computing fat-tree data center networks," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1156-1172, 2022.

[6] T. Feng, S. Miao, C. Liu, R. Ma, "Verifiable keyword search encryption scheme that supports revocation of attributes," *Symmetry*, vol. 15, no. 4, 2023.

[7] H. Gao, H. Huang, L. Xue, F. Xiao and Q. Li, "Blockchain-enabled fine-grained searchable encryption with cloud-edge computing for electronic health records sharing," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18414-18425, 2023.

[8] N. Ghosh, S. K. Ghosh and S. K. Das, "SelCSP: A framework to facilitate selection of cloud service providers," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 66-79, 2015.

[9] H. Guo, X. Yu, "A survey on blockchain technology and its security," *Blockchain: research and applications*, vol. 3, no. 2, 2022.

[10] K. Guo, Y. Han, R. Wu, K. Liu, "CD-ABSE: Attribute-based searchable encryption scheme supporting cross-domain sharing on blockchain,"

Wireless Communications and Mobile Computing, vol. 2022, 2022.

[11] Y. Guo, C. Zhang, C. Wang and X. Jia, "Towards public verifiable and forward-privacy encrypted search by using blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2111-2126, 2023.

[12] Y. Jiang, X. Xu and F. Xiao, "Attribute-based encryption with blockchain protection scheme for electronic health records," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3884-3895, Dec. 2022.

[13] Y. Li, F. Zhou, D. Ji, Z. Xu, "A hierarchical searchable encryption scheme using blockchain-based indexing," *Electronics*, vol. 11, no. 22, pp. 3832, 2022.

[14] M. Mihailescu, S. Nita, "Searchable encryption," in *Pro Cryptography and Cryptanalysis with C++20. Apress, Berkeley, CA*, 2021. https://doi.org/10.1007/978-1-4842-6586-4_11

[15] S. Niu, M. Song, L. Fang, F. Yu, S. Han, C. Wang, "Keyword search over encrypted cloud data based on blockchain in smart medical applications," *Computer Communications*, vol. 192, pp. 33-47, 2022.

[16] M. S. Rahman, M. Chamikara, I. Khalil, "Blockchain-of-blockchains: An interoperable blockchain platform for ensuring IoT data integrity in smart city," *Journal of Industrial Information Integration*, vol. 30, 2022.

[17] M. Ramachandran, V. Chang, "Towards performance evaluation of cloud service providers for cloud data security," *International Journal of Information Management*, vol. 36, no. 4, pp. 618-625, 2016.

[18] S. Ramzan, A. Aqdus, V. Ravi, D. Koundal, R. Amin and M. A. Al Ghamdi, "Healthcare applications using blockchain technology: Motivations and challenges," *IEEE Transactions on Engineering Management*, vol. 70, no. 8, pp. 2874-2890, 2023.

[19] Y. S. Rao, S. Prasad, S. Bera, A. K. Das and W. Susilo, "Boolean searchable attribute-based signcryption with search results self-verifiability mechanism for data storage and retrieval in clouds," *IEEE Transactions on Services Computing*, doi: 10.1109/TSC.2023.3327816.

[20] S. Sarkodie, M. Ahmed, T. Leirvik, "Trade volume affects bitcoin energy consumption and carbon footprint," *Finance Research Letters*, vol. 48, 2022.

[21] J. Su, L. Zhang, Y. Mu, "BA-RMKABSE: Blockchain-aided ranked multi-keyword attribute-based searchable encryption with hiding policy for smart health system," *Future Generation Computer Systems*, vol. 132, pp. 299-309, 2022.

[22] Q. Tong, Y. Miao, J. Weng, X. Liu, K. -K. R. Choo and R. H. Deng, "Verifiable fuzzy multi-keyword search over encrypted data with adaptive security," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 5386-5399, 2023.

[23] X. Wang, S. Yin, H. Li, L. Teng, S. Karim, "A modified homomorphic encryption method for multiple keywords retrieval," *International Journal of Network Security*, vol. 22, no. 6, pp. 905-910, 2020.

[24] X. Wang, S. Yin, M. Shafiq, A. A. Laghari, S. Karim, O. Cheikhrouhou, W. Alhakami, H. Hamam, "A new v-net convolutional neural network based on four-dimensional hyperchaotic system for medical image encryption," *Security and Communication Networks*, vol. 2022, 2022. https://doi.org/10.1155/2022/4260804

[25] N. Wu, L. Xu, L. Zhu, "A blockchain based access control scheme with hidden policy and attribute," *Future Generation Computer Systems*, vol. 141, pp. 186-196, 2023.

[26] C. Xu, P. Zhang, L. Mei, Y. Zhao, L. Xu, "Ranked searchable encryption based on differential privacy and blockchain," *Wireless Networks*, pp. 1-14, 2022.

[27] P. Xu, J. Chen, Y. Yang and J. Ning, "DuMSE: Toward practical and dynamic multiuser search over encrypted cloud data against keyword guessing attack," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1082-1095, 2023.

[28] Z. Xu, S. Zhang, H. Han, X. Dong, Z. Zheng, H. Wang, W. Tian, "Blockchain-aided searchable encryption-based two-way attribute access control research," *Security and Communication Networks*, vol. 2022, 2022.

[29] L. Yan, L. Ge, Z. Wang, G. Zhang, J. Xu, Z. Hu, "Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1-16, 2023.

[30] S. Yin, H. Li, L. Teng, A. Laghari, V. V. Estrela, "Attribute-based multiparty searchable encryption model for privacy protection of text data," *Multimedia Tools and Applications*, 2023. https://doi.org/10.1007/s11042-023-16818-4

[31] S. Yin, J. Liu, L. Teng, "A sequential cipher algorithm based on feedback discrete hopfield neural network and logistic chaotic sequence," *International Journal of Network Security*, vol. 22, no. 5, pp. 869-873, 2020.

[32] H. Zeng, H. Zamani, V. Vinay, "Curriculum learning for dense retrieval distillation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1979-1983, 2022.

[33] F. Zhang, Y. Zhang, G. Han, "Blockchain-based attribute-based keyword searchable encryption for health cloud system," *International Journal of Embedded Systems*, vol. 15, no. 6, pp. 493-504, 2022.

Biography

Hang Li biography. He obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hang Li is a full professor of the software college at Shenyang Normal University. His interests are wireless networks, mobile computing, cloud computing,

social networks, network security and quantum cryptography. Prof. Li had published more than 30 international journal and international conference papers on the above research fields. Email:lihangsoft@163.com.

Zeyang Li biography. Zeyang Li is with the Software College, Shenyang Normal University. His major is computer science, information secure.

Xiaowei Wang biography. She is a full professor of the software college at Shenyang Normal University. Her interests are wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Prof. Wang had published more than 10 international journal and international conference papers on the above research fields. Email:hsiaoweiw@163.com

Muhammad Ibrar biography. Muhammad Ibrar is with the Software College, Shenyang Normal University. His major is computer science, information secure.

Xinjie Zhu is with the Zhengzhou University of Science and Technology. Several papers had been puiblished related to the major. Research interest is: education data analysis, information processing.