Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3265934, 10 pages
https://doi.org/10.1155/2021/3265934

Research Article

Research on the Ranked Searchable Encryption Scheme Based on an Access Tree in IoTs

Yan-Yan Yang,[1] Bei Gong,[1,2] Zhi-Juan Jia,[1,3] Ya-Ge Cheng,[1] and Yu-Chu He[1]

[1]*School of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450044, China*
[2]*College of Computer Science, Beijing University of Technology, Beijing 100124, China*
[3]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Bei Gong; gongbei@bjut.edu.cn

Received 6 August 2021; Accepted 13 September 2021; Published 31 October 2021

Academic Editor: Zhihan Lv

With the continuous development of the Internet of things (IoTs), data security and privacy protection in the IoTs are becoming increasingly important. Aiming at the hugeness, redundancy, and heterogeneity of data in the IoTs, this paper proposes a ranked searchable encryption scheme based on an access tree. First, this solution introduces parameters such as the word position and word span into the calculation of the relevance score of keywords to build a more accurate document index. Secondly, this solution builds a semantic relationship graph based on mutual information to expand the query semantics, effectively improving the accuracy and recall rate during retrieval. Thirdly, the solution uses an access tree control structure to control user authority and realizes fine-grained access management to data by data owners in the IoTs. Finally, the safety analysis of this scheme and the efficiency comparison with other existing schemes are given.

1. Introduction

With the rapid development of IoTs technology, it has been used in all walks of life and has been widely recognized in various fields such as medical care, smart transportation, government work, smart home, and environmental monitoring [1–5]. At the same time, all kinds of information generated by users are increasing by a huge order of magnitude. Cloud storage is widely used due to its low cost and good scalability, and it solves the storage and management of this electronic data information by the IoTs. However, frequent privacy data leakage incidents have caused severe social impacts and disrupted economic development [6–8]. Therefore, how to protect user privacy and data security has become a technical bottleneck restricting the further development of IoT applications [9–13]. An effective way to solve data privacy leakage is to encrypt data and then store it on a cloud server. It can prevent unauthorized servers from accessing user data, and it can also effectively protect user data when the server is attacked. However, when users want to access their own data, because the cloud server stores encrypted data, these data no longer maintain the plaintext data structure before encryption, so the cloud server cannot effectively return the data searched by the user. In view of this situation, the easiest way is to download all encrypted data locally and then decrypt them one by one before searching. This method does not make full use of the computing power of the cloud server and wastes a lot of time and bandwidth power consumption, so that it cannot meet the actual needs of the cloud storage of the IoTs. Therefore, how to securely retrieve ciphertext data is an urgent problem to be solved in the IoTs [14].

The searchable encryption (SE) technology is a special encryption technology that can realize keyword ciphertext retrieval and ensure that attackers cannot obtain the keyword information queried by users through keyword ciphertext or search trapdoors [15]. At present, searchable encryption technologies mainly include symmetric searchable encryption (SSE) and asymmetric searchable encryption (ASE) [16, 17]. In 2000, Song [18] proposed a single-keyword searchable encryption scheme based on symmetric key encryption, which searches the ciphertext of related keywords by linearly scanning the entire ciphertext document. In 2004, Dan et al. [19] proposed an asymmetric searchable encryption scheme (Public-Key Encryption with Keyword Search (PKES)) for mail routing application scenarios. After that, researchers have done a lot of research on this basis.

In practical applications, users are usually more concerned with finding the top K documents most relevant to multiple keywords. In order to meet such demands, various multikeyword ranking searchable encryption schemes have been proposed in recent years. In 2011, Cao et al. [20] based on the secure KNN technology [21] first proposed a multikeyword ranking searchable encryption scheme based on vector inner product calculation. The scheme uses a 0/1 vector to represent each document and query vector. Compare the number of digits in the same position with a value of 1 to obtain the relevance score of the document, but this solution does not consider the importance of different keywords in the document. Therefore, Sun et al. [22] extended the scheme, introduced keyword weights when constructing document vectors and query vectors, and calculated the correlation through vector cosines to improve the accuracy of ranking.

The above ranking searchable encryption schemes all focus on the precise search of keywords and do not take into account the semantic expansion of keywords, resulting in many documents that meet the query conditions not being retrieved. Yang et al. [23] proposed a fast multikeyword semantic ranking search scheme, which introduced the concept of domain weighted scoring into document scoring and semantically expanded search keywords to improve the accuracy of the document index. In addition, the document vector is divided into blocks to effectively filter a large number of irrelevant documents, which effectively improves the efficiency of the scheme. However, this solution does not involve access control and can only be limited to a single legitimate user's query and is not suitable for the needs of keywords being queried by multiple users in an actual environment. Sun et al. [24] proposed an attribute-based keyword search scheme, which only returns authorized documents to search users. However, the search results cannot be ranked. Li et al. [25] proposed an authorized multikeyword ranking search scheme based on encrypted cloud data using attribute-based encryption strategy and symmetric searchable encryption. This scheme satisfies the confidentiality of files, the unlinkability of trapdoors, and the resistance to collusion attacks. Moreover, the scheme can enable the same data to be queried by multiple users but does not consider the semantic relevance of search keywords.

Therefore, the research on searchable encryption schemes for the cloud storage environment of the IoTs not only must protect the privacy of data to achieve the purpose of secret search but also ensure the efficiency of search efficiency. At the same time, it is also necessary to consider the situation of multiple users accessing search in the special application scenario of the IoT cloud storage environment. In the solution, the access tree is used to set user access permissions, which allows only authorized users to retrieve cloud data and obtain the most relevant K documents. And based on the secure KNN method, the document is encrypted to ensure the security and correctness of index creation and trapdoor generation.

The main contributions of this paper are as follows:

(1) This paper introduces parameters such as word position and word span into the calculation of the relevance score of keywords and assigns more accurate weights to keywords at different positions in the document, thereby constructing a more accurate document index

(2) This paper builds a semantic relationship graph based on mutual information to expand the query vector semantics, which effectively improves the precision and recall during retrieval

(3) This paper involves multikeyword search and access control. The access tree is used to control user access rights. Only users whose attributes meet the access policy defined by the data owner can search encrypted data with multiple keywords, so as to realize the fine-grained access management of the data by the data owner in the IoTs

2. Preliminaries

2.1. Vector Space Model. The vector space model [23] is the representation of the document set in the same vector space. Each document corresponds to a document vector, the dimension of the vector is equal to the length of the keyword collection, each dimension of the vector corresponds to a keyword and the value is equal to the weight of the corresponding keyword in that dimension. The user's query is also regarded as a vector in the same space, which is called the query vector. The keywords corresponding to each dimension of the vector are consistent with the document vector, and the vector dimension is the same as the document vector. The relevance score of the query and each file is equal to the value of the inner product of the document vector and the query vector.

2.2. Word Span. Word span [19] refers to the distance between the first and last occurrence of a word or phrase in the document. The larger the word span, the more important the word is to the topic of the document. The word span can effectively reduce the impact of local keywords on document keyword extraction, because local keywords often become keywords in the entire document due to their high-frequency advantages, reducing the accuracy of keyword extraction. The word span formula is shown in formula (1).

$$\mathrm{Span}_{ij} = \frac{\mathrm{last}_{ij} - \mathrm{first}_{ij} + 1}{\mathrm{sum}_{ij}}. \tag{1}$$

Among them, first_{ij} is the location identifier where the keyword w_j first appeared in the document f_i, last_{ij} is the location identifier where the keyword w_j last appeared in the document f_i, and sum_{ij} is the total number of keywords in the document f_i obtained after word segmentation processing.

2.3. Word Position. The word position [26] refers to the area where a keyword appears in a document, which is of great value for judging the importance of the keyword. The title and abstract are the central ideas extracted by the author through the summary of the whole article, so the keywords appearing in these two positions are more important than those appearing in the main text. This article divides the word position into three parts: title, abstract (or first paragraph), and body. Here, let the position value area_{ij} of the keyword w_j in different areas of the document f_i be set to 3, 2, and 1. There are two situations where a keyword appears multiple times:

(1) If the same area appears multiple times, the record is not repeated

(2) If it appears multiple times in different areas, the highest value is used

The word position formula is shown in formula (2).

$$\text{loc}_{ij} = \frac{\text{area}_{ij} - 1}{\text{area}_{ij} + 1}. \tag{2}$$

2.4. Relevance Score. This paper is based on the calculation method of TF-IDF (term frequency-inverse document frequency) to evaluate the importance of keywords in documents.TF represents the frequency of the keyword appearing in the document, and IDF represents the frequency of the inverse document, that is, the fewer the documents containing the keyword, the greater the IDF value, indicating that the keyword has a strong distinguishing ability. The TF-IDF formula is shown in formula (3).

$$\text{tf} - \text{idf}_{ij} = \text{TF} \times \text{IDF} = \text{tf}_{ij} \times \log\left(\frac{N}{n_j}\right), \tag{3}$$

where tf_{ij} represents the frequency of the keyword w_j in the document f_i, N represents the total number of all documents, and n_j represents the number of documents containing the keyword w_j.

When calculating the relevance score of keywords, the word position and word span factors of the keywords should also be considered.

Therefore, the correlation score formula used in this paper is shown in formula (4).

$$\text{Score}_{ij} = \alpha \times \text{tf} - \text{idf}_{ij} + \beta \times \text{span}_{ij} + \gamma \times \text{loc}_{ij}. \tag{4}$$

Among them, α, β, γ represent the weight of the three parameters and $\alpha + \beta + \gamma = 1$.

2.5. Semantic Relation Graph. Mutual information allows users to analyze the correlation between keywords. Constructing a semantic relationship graph [27] based on mutual information to expand the query semantics can effectively improve the precision and recall during retrieval. For keywords x and y, their mutual information $I(x, y)$ [28] is expressed as shown in formula (5).

$$I(x, y) = lb\frac{p(x, y)}{p(x)p(y)}, \tag{5}$$

where $p(x)$ represents the probability of a document, and $p(x, y)$ represents the probability of a document containing both x and y. Then, normalize the information.

$$I(x, y) \longleftarrow \frac{I(x, y)}{I_{\max}}, \tag{6}$$

where $I_{\max}$ represents the maximum mutual information value in all $I(x, y)$. Figure 1 is a small-scale semantic relationship graph $G(V, E)$, where node v represents the keyword and the edge weight e_{ij} represents the normalized mutual information value of two related keywords v_i, v_j.

2.6. Access Tree. The scheme in this paper uses the access tree defined by the CP-ABE [29] scheme to represent the access structure. The access tree can be flexibly and efficiently applied to access authority control, which is defined as follows.

Let Υ denote the visit tree, and each nonleaf node in Υ represents a threshold. If node x has num_x child nodes and its threshold is k_x, then, $0 < k_x \leq \text{num}_x$. When $k_x = 1$, the node represents an OR gate. If $k_x = \text{num}_x$, it means the AND gate. Each leaf node in Υ represents an attribute and the leaf node corresponds to $k_x = 1$.

When checking whether the user authority meets the access tree Υ, let R be the root node of Υ and let Υ_x be the subtree with node x as the root node. If the attribute set Att can satisfy the strategy represented by Υ_x, then, denote $\Upsilon_x(\text{Att}) = 1$. Calculate $\Upsilon_x(\text{Att})$ using the following recursive algorithm.

If x is a nonleaf node, then, calculate $\Upsilon_x{}'(\text{Att})$ for the child node x' of x. Only when the number of child nodes satisfying $\Upsilon_x{}'(\text{Att}) = 1$ is greater than or equal to k_x, then, let $\Upsilon_x(\text{Att}) = 1$; otherwise, it is NULL. If the node is a leaf node, only if the corresponding attribute of the node is $\text{attr}(x) \in \text{Att}$, then, let $\Upsilon_x(\text{Att}) = 1$; otherwise, it is NULL.

2.7. Bilinear Mapping. G, G_T are two multiplicative cyclic group of prime order p, g is a generator of G, $e : G \times G \longrightarrow G_T$ is a bilinear map [30] if three properties are satisfied:

(1) Bilinear. For $a, b \in Z_p$ and $\forall g_1, g_2 \in G$, $e(g_1^a, g_2^b) = e(g_1, g)_2^{ab}$

(2) Nondegenerate. $e(g_1, g_2) \neq 1$

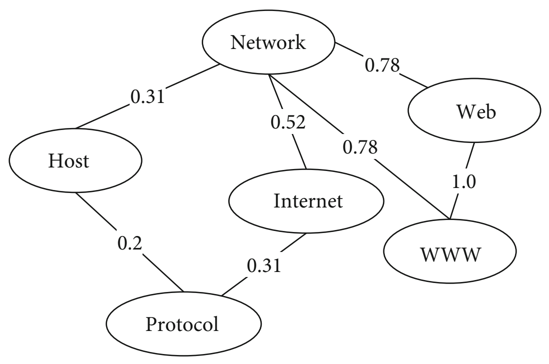

Figure 1: Semantic relationship graph between keywords.

(3) Computability. There is an efficient algorithm computing $e(g_1, g_2)$, for any $g_1, g_2 \in G$.

It is said that e is an effective bilinear mapping from G to G_T.

3. Problem Description

3.1. System Model. The entities included in this program include data owners (Data Owner (DO)), data users (Data User (DU)), IoT cloud servers (IoT Cloud Server (CS)), and trusted institutions (Trust Authority (TA)). The system model is shown in Figure 2.

(1) Data Owner. The data owner is responsible for encrypting the original document, establishing a secure index and uploading the ciphertext document and the secure index. First, the data owner extracts the keyword collection from the original document collection and encrypts it according to the keyword collection and the data access strategy to generate a security index and then uses the symmetric key to encrypt the original document collection to generate a ciphertext document collection. Finally, the ciphertext document collection and the security index are uploaded and stored to the cloud server together.

(2) IoT Cloud Server. The IoT cloud server is mainly responsible for receiving and storing the data uploaded by the data owner and satisfying the query requests of authorized users. When receiving a user's query request, the cloud server first conducts permission review. If it is an authorized user, use the stored security index and trapdoor to calculate the similarity score of the document, search for related documents, sort the query results, and return the most relevant TOP-K document to the user. It is worth noting that only authorized users can perform a correct search and unauthorized users cannot obtain search results.

(3) Trust Authority. It is mainly responsible for generating system keys and generating user private keys based on user attribute sets.

(4) Data User. The data user submits a query request to the cloud storage server of the IoTs to query the files of interest. The user sends his own set of attributes to a trusted organization to obtain the user's private key and then uses the private key and query keywords to generate trapdoors and permission tags and upload them to the cloud server. Finally, authorized users can receive the most relevant TOP-K query results sent by the cloud server.

3.2. Safety Requirements. This paper assumes that trust authority is completely credible. 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, and it may try to obtain other additional information from the security index and trapdoor. Therefore, the solution in this paper mainly considers the following 4 types of security requirements:

(1) Confidentiality of Documents. The data owner does not want unauthorized entities (cloud servers or data users) to know the content of the documents, so the documents must be encrypted before they are sent to the cloud servers and the unauthorized entities do not have decryption keys.

(2) Anonymity of Indexes and Trapdoors. The cloud server knows the ciphertext information stored by the data owner, including ciphertext document collection, security indexes, and trapdoors, but does not know the key.

(3) Anonymous Access. Data users can access IoT data without giving their detailed identity information.

(4) Collusion Resistance. Any two or more data users cannot collude to access the document.

3.3. Scheme Definition. The multikeyword semantic rank search scheme based on the user attribute consists of 5 polynomial time algorithms such as Setup, Encrypt, KeyGen, Trapdoor, and Search:

(1) Setup(1^κ). TA runs the initialization algorithm and generates system master key MSK, index key IK, and system public parameter PK by inputting system security parameter κ.

(2) KeyGen(MSK, Att). This algorithm is the user's private key generation algorithm, which is executed by TA. The algorithm inputs the system master key MSK and user attribute set Att and outputs the user private key SK.

(3) Encrypt(IK, PK, FF, Υ). The data owner executes the encryption algorithm. The algorithm inputs the index key IK, the system public parameters PK, the plain text document collection FF, and the access tree Υ and outputs the security index I and the cipher text document collection CC.

(4) Trapdoor(W_Q, IK, PK, SK). The data user uses the algorithm to generate search credentials corresponding to the keywords that need to be queried. The algorithm inputs

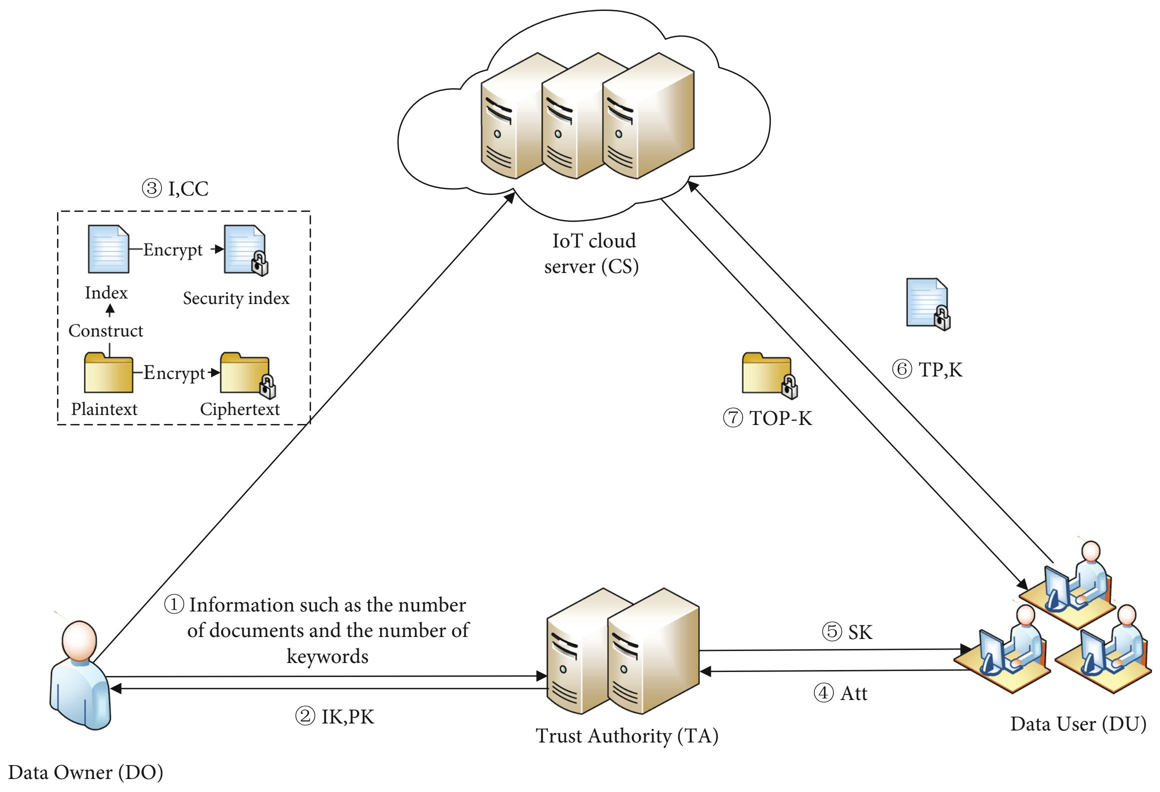

FIGURE 2: System model.

the query keyword set W_Q, index key IK, system public parameters PK, and user private key SK and outputs search credentials TP.

(5) Search(I, TD, K). The keyword search algorithm is executed by the cloud server. The algorithm inputs the security index I, search credentials TP, and the parameter K and outputs the TOP-K documents most relevant to the query keyword set. It is worth noting that only users who meet the access control authority can get the correct results; otherwise, the search will fail.

3.4. Scheme Description

(1) Setup$(1^\kappa) \longrightarrow \{\mathrm{MSK}, \mathrm{IK}, \mathrm{PK}\}$. TA randomly selects a large prime number $p (p \in Z_p)$. Let G, G_T be the multiplicative cyclic group whose generator are g and the order are p. TA generates a bilinear map $e : G \times G \longrightarrow G_T$ and a hash function $H_1 : \{0, 1\}^* \longrightarrow G$. In addition, TA randomly generates an $m + \varepsilon$-dimensional segmentation vector S and two $(m + \varepsilon) \times (m + \varepsilon)$-dimensional invertible matrices $\{M_1, M_2\}$, where ε is the number of confusion bits and m is the number of keywords, and generate index key $\mathrm{IK} = \{S, M_1, M_2\}$. Finally, TA randomly selects $\alpha, \beta \in Z_p$ and generates system master key $\mathrm{MSK} = \{\alpha, \beta\}$ and system public parameters $PK = \{g, G, G_T, e(g, g), e(g, g)^\alpha, H_1, g^\alpha, g^\beta\}$.

(2) KeyGen$(\mathrm{MSK}, \mathrm{Att}) \longrightarrow \mathrm{SK}$. TA selects a random number $r \in Z_p$ and randomly selects $r_j \in Z_p$ for each attribute a_j in the attribute set Att and finally generates the user's private key $\mathrm{SK} = \{K = g^{(\alpha+r)/\beta}, \forall a_j \in \mathrm{Att} : D_j = g^r H_1(a_j)^{r_j}, D'_j = g^{r_j}\}$. The system transfers the user's private key SK to the data user.

(3) Encrypt$(\mathrm{IK}, \mathrm{PK}, \mathrm{FF}, \Upsilon) \longrightarrow \{I, CC\}$.

(1) Extract Keywords.

The data owner extracts keywords from the plaintext document collection $\mathrm{FF} = \{f_1, f_2, \cdots, f_m\}$ to obtain the keyword collection $W = \{w_1, w_2, \cdots, w_n\}$.

(2) Encrypted Documents.

The data owner uses the symmetric key ek to encrypt each document to obtain the ciphertext set $\mathrm{CC} = \{c_1, c_2, \cdots, c_m\}$.

(3) Create Index Process.

Based on the vector space model, the data owner generates a document vector D_i for each document f_i. If the document contains the keyword w_j, use formula (4) to calculate the relevance score $D_i[j] = \mathrm{score}_{ij}$ of the keyword w_j in the document. Otherwise, $D_i[j] = 0$.

The data owner expands each document vector D_i from the m dimension to the $m + \varepsilon$ dimension and sets $D_i[m + t] = \eta_t$, where $1 \leq t \leq \varepsilon$ and η_t are random numbers with normal distribution $N(\mu, \sigma^2)$. Then, the data owner splits each document D_i into two vectors $\{D'_i, D''_i\}$ according to the split vector S. If $S[j]=0, j=1,2,\cdots,m+\varepsilon$, then, $D_i'[j]=D_i''[j]=D_i[j]$, if $S[j]=1$, let $D_i'[j]$ and $D_i''[j]$ be random values and $D_i'[j]+D_i''[j]=D_i[j]$. Finally, the data owner uses the master key MSK to encrypt $D_i'[j]$ and $D_i''[j]$ and get the partial security index $I_i=\{M_1^T D_i', M_2^T D_i''\}$.

According to the access tree Υ, a polynomial q_x is selected for each node x in Υ and the polynomial q_x is generated as follows. Starting from the root node R of Υ, use a recursive algorithm to run from top to bottom. For each node x, let the number of terms d_x of the polynomial q_x be one less than the threshold k_x represented by the node, that is, $d_x=k_x-1$. First, select $s\in Z_p$ randomly for the root node R and let $q_R(0)=s$, and then, randomly select the coefficients of other terms. For other nodes x, define the function parent(x), index(x), the former represents the parent node of node x, and the latter represents the position of node x in the parent node. Let $q_x(0)=q_{\text{parent}(x)}(\text{index}(x))$, and randomly select coefficients for the other terms of q_x. According to the above algorithm, C_v, C_v' is generated for all nodes in Υ, namely, $I_\Upsilon=\{C=\text{ek}\cdot e(g,g)^{\alpha s}, W_1=g^{\beta s}, \forall v\in\Upsilon, C_v=g^{q_v(0)}, C_v'=H_1(\text{attr}(v))^{q_v(0)}\}$. Finally, a safety index $I=\{I_i, I_\Upsilon\}$ is generated.

Finally, the data owner uploads the security index I and the ciphertext document collection CC to the cloud server.

(4) Trapdoor(W_Q, IK, PK, SK) $\longrightarrow$ TP.

(1) Keyword Semantic Expansion.

First, the data user performs semantic expansion on the keyword set W_Q according to the semantic relationship graph to obtain the expanded keyword set W_Q'.

(2) Generate the Query Vector.

Based on the vector space model, the query vector Q is constructed. Here, if $w_j\in W_Q$, then let $Q[j]=1$; if the expansion word w_j corresponds to one original keyword w_i, then, $Q[j]=e_{ij}$; similarly, if the expansion word w_j corresponds to multiple original keywords w_i, then, $Q[j]=\{e_{ij}\}_{\max}$. Finally, extend the query vector Q from the m dimension to the $m+\varepsilon$ dimension and let $Q[m+t]=\tau_t$, where τ_t is a random number and $1\le t\le\varepsilon$.

(3) Encrypt the Query Vector.

First, the data user divides the query vector Q into two vectors $\{Q', Q''\}$ according to the division vector S. This is the opposite of the document D_i split method. If $S[j]=1, j=1,2,\cdots,m+\varepsilon$, then, $Q'[j]=Q''[j]=Q[j]$; if $S[j]=0$, let $Q'[j]$ and $Q''[j]$ be random values and $Q'[j]+Q''[j]=Q[j]$. Finally, the data user encrypts $Q'[j]$ and $Q''[j]$ with the system master key MSK to get the trapdoor $\text{TD}=\{M_1^{-1}Q', M_2^{-1}Q''\}$. Then, randomly select $\theta\in Z_p$ and generate search credentials $\text{TP}=\{\text{TD}, T_1=K^\theta=g^{(\alpha+r)\theta/\beta}, \forall a_j\in\text{Att}: E_j=D_j^\theta=g^{\theta r}H_1(a_j)^{\theta r_j}, E_j'=D_j'^\theta=g^{r_j\theta}\}$.

Finally, the data user sends the search credentials TP to the cloud server.

(5) Search(I, TP, K) $\longrightarrow C_{\text{TOP-K}}$.

If the cloud server receives the query request from the data user, it can perform the following steps:

First, the cloud server first calculates whether the user attributes meet the access tree defined by the data owner. x is a node in the access tree Υ, and the cloud server executes the following recursive algorithm:

If node x is a leaf node, let $a_j=\text{attr}(x)$ be the attribute corresponding to node x. If $a_j\in\text{Att}$, then,

$$\text{DecryptNode}(x)=\frac{e(E_j,C_x)}{e(E_j',C_x')}=\frac{e\left(g^{\theta r}H_1(a_j)^{\theta r_j}, g^{q_x(0)}\right)}{e\left(g^{r_j\theta}, H_1(a_j))^{q_x(0)}\right)}=e(g,g)^{\theta r q_v(0)}. \tag{7}$$

Otherwise, DecryptNode(x) = NULL.

If node x is a nonleaf node, calculate $F_z=\text{DecryptNode}(z)$ for the child node z of node x. Let S_x be the set of k_x subnodes that satisfy $F_z\neq\text{NULL}$. If no such set S_x, namely, DecryptNode(x) = NULL, is found, it means that the access requirements are not met. Otherwise, calculate

$$\begin{aligned}F_x&=\prod_{z\in S_x}F_z^{\Delta_{j,S_x'}(0)}=\prod_{z\in S_x}\left(e(g,g)^{\theta r q_z(0)}\right)^{\Delta_{j,S_x'}(0)}=\prod_{z\in S_x}\left(e(g,g)^{\theta r q_{\text{parent}(z)}(\text{index}(z))}\right)^{\Delta_{j,S_x'}(0)}\\&=\prod_{z\in S_x}\left(e(g,g)^{\theta r q_x(j)}\right)^{\Delta_{j,S_x'}(0)}=e(g,g)^{\theta r q_x(0)}.\end{aligned} \tag{8}$$

Using the Lagrangian interpolation theorem, $V=\text{DecryptNode}(R)=e(g,g)^{\theta rs}$ can be obtained. Here, it is explained that the data user is an authorized user who can perform data query.

The cloud server uses formula (9) to calculate the correlation between the security index I_i and the query trapdoor TD and returns the TOP-K documents most relevant to the query keyword set to authorized users. If the user does not meet the access rights, the search fails and NULL is returned.

The formula for calculating document relevance is shown in formula (9).

$$\begin{aligned}I_i\cdot\text{TD}&=\left\{M_1^T D_i', M_2^T D_i''\right\}\cdot\left\{M_1^{-1}Q', M_2^{-1}Q''\right\}\\&=\left(M_1^T D_i'\right)\cdot\left(M_1^{-1}Q'\right)+\left(M_2^T D_i''\right)\cdot\left(M_2^{-1}Q''\right)\\&=\left(M_1^T D_i'\right)^T\left(M_1^{-1}Q'\right)+\left(M_2^T D_i''\right)^T\left(M_2^{-1}Q''\right)\\&=D_i'^T M_1 M_1^{-1}Q'+D_i''^T M_2 M_2^{-1}Q''=D_i'^T Q'+D_i''^T Q''\\&=D_i'\cdot Q'+D_i''\cdot Q''=D_i\cdot Q.\end{aligned} \tag{9}$$

If the user's attribute set Att satisfies part or all of Υ, the user can obtain $V = e(g, g)^{\theta rs}$ according to formula (8) and calculate the document encryption key by formulas (10) and (11).

$$V_1 = \frac{e(W_1, T_1)}{V} = \frac{e\left(g^{\beta s}, g^{(\alpha+r)\theta/\beta}\right)}{e(g, g)^{\theta rs}} = e(g, g)^{s\alpha\theta}, \tag{10}$$

$$\frac{C}{V_1^{1/\theta}} = \frac{\text{ek} \bullet e(g, g)^{\alpha s}}{\left(e(g, g)^{s\alpha\theta}\right)^{1/\theta}} = \text{ek}. \tag{11}$$

Finally, the user uses ek to decrypt the obtained document to obtain a collection of plaintext documents.

4. Safety Analysis

4.1. Confidentiality of Documents. The document is encrypted with a symmetric key before being uploaded to the cloud server, and only data users who meet the access policy defined by the data owner can search for the document and further obtain the decryption key to decrypt the obtained ciphertext document. Therefore, this solution guarantees the confidentiality of the document.

4.2. Anonymity of Indexes and Trapdoors. The cloud server can get the encrypted security index I and query vector Q. The security index I_u of each document is represented as $\{M_1^T D'_u, M_2^T D'_u\}$ under the action of the segmentation vector and the invertible matrix, namely, $\text{Index}_u = \begin{cases} C_1 = M_1^T D'_u \\ C_2 = M_2^T D'_u \end{cases}$, where the segmentation vector S and the two invertible matrices M_1, M_2 are the encryption keys of this scheme. It can be seen from the foregoing that in the above equations, M_1, M_2, D'_u, and D'_u are all $(m+\varepsilon)$-dimensional vectors (here, ε is equal to 0), so there are $2m$, n equations in a set containing n documents. However, there are $2m^2$ unknowns in M_1, M_2, and $2mn$ unknowns in D'_u,D'_u. It is not feasible to solve such a system of equations in which the number of equations is less than the number of unknowns, so the cloud server cannot deduce M_1, M_2, D'_u, and D'_u.

Similarly, the query vector Q can be regarded as two m-dimensional vectors $\{Q', Q''\}$, that is, the number of unknowns is $2m$. There are $2m^2$ unknowns in M_1,M_2. However, the number of equations for solving the query vector is only $2m$, so the query vector Q and the invertible matrix M_1, M_2 cannot be solved as well. Therefore, this scheme can ensure the safety of indexes and query vectors.

4.3. Anonymous Access. The solution uses attributes as the minimum granularity of access control. When an access request is made, the IoTs does not care about the user's identity and only needs to verify whether the user's attributes meet the access structure and decide whether to provide the user with decrypted data.

4.4. Collusion Resistance. Collusion resistance means that users with different attributes cannot decrypt the corresponding ciphertext even if they combine their private keys. In the searchable encryption scheme, it is required that even if users collude, they cannot search for unauthorized keyword ciphertexts. In this scheme, the system selects a random number $r \in Z_p$ for each attribute on the access tree. Since r is randomly distributed, the private keys of the same attribute in different networks are different, so that the secret value $e(g, g)^{r\theta s}$ that can be recovered is different. Therefore, this scheme has the property of anticollusion.

4.5. Function and Safety Comparison. In this section, we compare the expression ability and supported functions of the proposed scheme with some existing schemes. The summary is shown in Table 1.

5. Efficiency Analysis

The following analyzes the computational cost performance of this scheme from the stages of private key generation, indexing, trapdoor, search, etc. and compares the efficiency of the scheme in the literature [31] with the scheme in this paper and then conducts an experimental simulation on the scheme, and the following situations can be ignored.

(1) Index Generation Stage. For encrypting each document index, the data owner performs the multiplication of two ($m+\varepsilon$)-dimensional vector and $(m+\varepsilon) \times (m+\varepsilon)$-dimensional matrix with a complexity of $O((m+\varepsilon)^2)$, where $m+\varepsilon$ is the number of keywords after expansion. Comparing the exponential operation and bilinear pairing operation on the group G, G_T, the time spent on the matrix multiplication operation is negligible.

(2) The Trapdoor Generation Stage. For calculating the encrypted query vector, the data user needs to perform the multiplication operation between the two $(m+\varepsilon)$-dimensional vector and the $(m+\varepsilon) \times (m+\varepsilon)$-dimensional matrix and the time spent in the multiplication operation can also be ignored.

(3) Search Stage. If the user meets the access rights, the cloud server performs a search. The main operation is the inner product operation of two $(m+\varepsilon)$-dimensional vectors. The computational complexity is $O(m(m+\varepsilon))$, where m is the number of the entire document collection. Also here, the time spent in the vector inner product operation can be ignored.

Here, let T_g and T_{gt} denote the exponential operation of groups G and G_T, respectively, T_p denote bilinear pairing operation, T_h denote the time of hash operation, n is the number of attributes in the system, s is the number of user attributes, $|F|$ is the number of files, and $|W|$ is the number of keywords.

The efficiency comparison of the scheme in literature [31] and the scheme in this paper is shown in Table 2.

In order to verify the effectiveness of the scheme, this paper compares the performance of the scheme in literature [31] with the performance of this scheme. We conduct real

Table 1: Functional analysis comparison.

Features	MRSE [20]	Sun [24]	Yang [23]	Li[25]	This scheme
Access control	×	✓	×	✓	✓
Multiple keywords	✓	✓	✓	✓	✓
Semantic extension	×	×	✓	×	✓
Relevance ranking	✓	×	✓	✓	✓
Multi-user	×	✓	×	✓	✓

Table 2: Comparison of efficiency.

Algorithm	Literature [31]	This scheme
KenGen	$(2s+3)T_g + sT_h$	$(2s+1)T_g + sT_h$
Encrypt	$(2\|F\| + 2n + \|W\|)T_g + nT_h$	$(n+1)T_g + nT_h + T_{tg}$
Trapdoor	$(2s+4)T_g$	$(2s+1)T_g$
Search	$(2s+3)T_p + 2T_{gt}$	$(2s+1)T_p + 3T_{gt}$

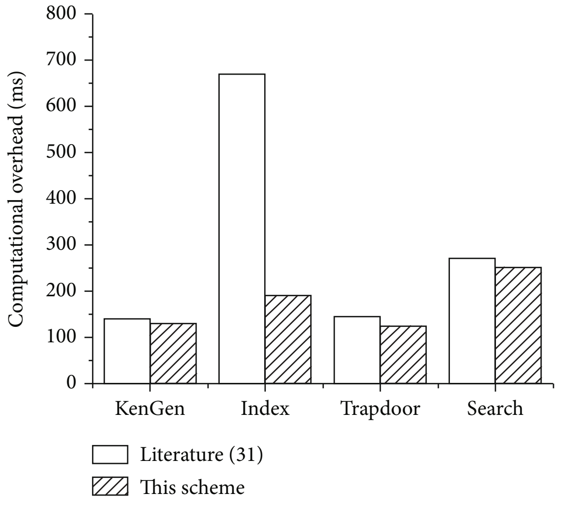

Figure 3: Computational overhead comparison.

experiments on a Windows 10 64-bit operating system, Inter (R) CoreTM i7-7700 CPU @ 3.60 GHz and 8 GB RAM to study the true execution time. Here, we set the number of keywords in the dictionary to be the same as the number of query keywords in the trapdoor ($|F| = |W|$) and set the number of attributes in the system to be equal to the number of user attributes ($n = s$), and $n = s = 10$, $|F| = |W| = 30$.

As shown in Figure 3, we found that compared with the computational cost in [31], in a large-scale data sharing system, the algorithm in this scheme is more computationally efficient, which means that this scheme is more effective and practical.

As shown in Figure 4, we compare the execution time of the search operation of one single subindex. The computational overhead of the search phase is mainly affected by the number of user attributes. We see that the computational overhead of the search phase of these two schemes increases linearly with the increase in the number of user attributes.

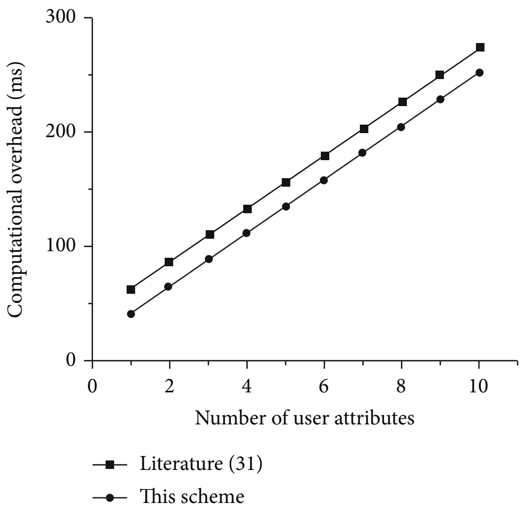

Figure 4: Search operation of one single subindex.

6. Conclusion

Aiming at the special application scenario of the IoTs environment, this paper proposes an attribute-based multikeyword ranking search scheme. The scheme not only realizes the keyword search function based on semantic expansion but also realizes the user's access control function. The scheme takes into account the weight difference of different positions of keywords and introduces parameters such as word position and word span into the calculation of the relevance score of keywords to build a more accurate document index. Secondly, the scheme expands the query keywords semantically according to the semantic relationship graph to find more keywords with similar meanings, thereby effectively improving the precision and recall rate during retrieval. Again, the solution uses an access tree control structure to control the access authority of data users and realizes the fine-grained management of data owners based on attributes. Finally, the functional and security analyses and comparison of the scheme show that the scheme has document confidentiality, index and trapdoor anonymity, anonymous access, and resistance to collusion attacks. In addition, the efficiency of the scheme is theoretically analyzed and the analysis results show that this scheme has advantages over other schemes.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study is supported by the National Key Research and Development Program of China (2020YFB1005404), Open Foundation of State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (SKLNST-2020-1-09), Henan Key Research Projects of Universities (21B520022, 20A580008), and Science and Technology Program of Henan Province (212102210415, 212102210100).

References

[1] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[2] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[3] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, 2021.

[4] J. W. Tang, R. X. Li, K. P. Wang, X. W. Gu, and Z. Y. Xu, "A novel hybrid method to analyze security vulnerabilities in android applications," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 589–603, 2020.

[5] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[6] Z. P. Cai and Z. B. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, Dallas, TX, USA, 2019.

[7] X. Zheng and Z. P. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[8] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable & Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[9] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, "Internet of things forensics: recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[10] L. Y. Qi, X. K. Wang, X. L. Xu, W. C. Dou, and S. C. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, 2021.

[11] Z. P. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. 7, no. 2, pp. 766–775, 2020.

[12] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[13] Y. Khazbak, J. Y. Fan, S. C. Zhu, and G. H. Cao, "Preserving personalized location privacy in ride-hailing service," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 743–757, 2020.

[14] Z. R. Shen, W. Xue, and J. W. Shu, "Survey on the research and development of searchable encryption schemes," *Journal of Software*, vol. 25, no. 4, pp. 880–895, 2014.

[15] X. L. Dong, J. Zhou, and Z. F. Cao, "Research advances on secure searchable encryption," *Journal of Computer Research an Development*, vol. 54, no. 10, pp. 2107–2120, 2017.

[16] H. Dai, X. Li, and X. Y. Zhu, "Research on multi-keyword ranked search over encrypted cloud data," *Computer Science*, vol. 46, no. 1, pp. 6–12, 2019.

[17] Y. Li and C. G. Ma, "Overview of searchable encryption research," *Chinese Journal of Network and Information Security*, vol. 4, no. 7, pp. 13–21, 2018.

[18] D. X. SONG, "Practical techniques for searches on encrypted data," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, USA, 2000.

[19] B. Dan, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-EUROCRYPT 2004, International conference on the theory and applications of cryptographic techniques*, Interlaken, Switzerland, 2004.

[20] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in *2011 Proceedings IEEE INFOCOM*, Shanghai, China, 2011.

[21] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009*, Providence, Rhode Island, USA, 2009.

[22] W. Sun, B. Wang, N. Cao et al., "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*, Hangzhou, China, 2013.

[23] Y. Yang, J. Liu, and S. W. Cai, "Fast multi-keyword semantic ranked search in cloud computing," *Computer Science*, vol. 41, no. 6, pp. 1126–1139, 2018.

[24] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2014.

[25] H. Li, D. Liu, K. Jia, and X. Lin, "Achieving authorized and ranked multi-keyword search over encrypted cloud data," in *IEEE International Conference on Communications*, London, UK, 2015.

[26] Y. J. Niu and C. L. Tian, "Research on TFIDF keyword extraction algorithm based on multiple factors," *Computer Technology and Development*, vol. 29, no. 7, pp. 80–83, 2019.

[27] X. Q. Pang, X. L. Yan, and W. J. Chen, "Dynamic multi-keyword ranked search over encrypted data supporting semantic extension," *Journal of Computer Applications*, vol. 39, no. 4, pp. 1059–1065, 2019.

[28] Z. Xia, L. Chen, X. Sun, and J. Liu, "A multi-keyword ranked search over encrypted cloud data supporting semantic extension," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 8, pp. 107–120, 2016.

[29] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, CA, USA, 2007.

[30] Y. Yang and M. Ma, “Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for E-health clouds,” *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 4, pp. 746–759, 2017.

[31] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, “Attribute-based keyword search over hierarchical data in cloud computing,” *IEEE Transactions on Services Computing*, vol. 13, pp. 985–998, 2017.