A Searchable Encryption Method Supporting Result Grouping and Sorting

HaiYang Peng[1,2,3(✉)], GuiShan Dong[2,3], Hao Yao[1,2,3], Yue Zhao[1,2,3], and YuXiang Chen[1,2,3]

[1] Science and Technology on Communication Security Laboratory, Chengdu 610041, China
physea@mail.ustc.edu.cn
[2] No. 30 Research Institute of China Electronics Technology Group Corporation, Chengdu 610041, China
[3] China Electronics Technology Cyber Security Co., Ltd, Chengdu 610041, China

Abstract. Aiming at the problem of low search efficiency caused by general searchable encryption schemes that do not sort search results during the search process, we propose a method to support grouping and sorting search results. This method uses the adjacent index structure as the basis, uploads the keywords corresponding to the file to the cloud server in order, and records the upload sequence of each keyword in the file, and stores the upload sequence in the index structure. The index is stored in the form of keyword-keyword sequence number-file address pointer. When the data user performs a keyword search, the keywords on the index are first matched, and then the results are grouped in the form of keyword serial number-file, and the search results are grouped and sorted according to the keyword serial number and returned to the user. The scheme proves the safety of the scheme through the method of bilinear difficulty analysis. At the same time, the scheme can support operations such as index update and deletion, and is easy to maintain.

Keywords: Searchable encryption · Packet sorting · Bilinear pairing

1 Introduction

With the rapid development of cloud computing and cloud storage technology, more and more enterprise and organizational users choose to store on cloud servers that can be flexibly purchased, allowing users to remotely access data stored in the cloud through the client. However, the data stored in the cloud in the clear state always has a large security risk, which means that the value of the data owned by the data owner is also outsourced to the cloud service provider. At the same time, the security of the cloud server is also limited by the security of the cloud service provider. Maintenance capabilities. Servers directly connected to the Internet will be subject to various attacks from the Internet. The series of data privacy breaches reported in recent years have caused great losses to users and greatly stimulated users' need for security. Stored in cloud storage.

S. Liu and X. Ma (Eds.): ADHIP 2021, LNICST 417, pp. 133–143, 2022.
https://doi.org/10.1007/978-3-030-94554-1_12

The cloud server is always semi-honest and will be interested in the data stored on the server. In order to avoid the curious cloud server from reading the data stored on it, the data can be stored in the cloud server in a secret storage method. Effectively prevent the cloud server from viewing the data stored on it, because only the user with the key can decrypt and view the stored data. At the same time, in order to enable data to be searched in a secret environment, searchable encryption technology [1] was proposed. Searchable encryption technology is a technology that can search for keywords in a secret environment and return the corresponding files to the user. However, early Searchable encryption technology only supports single-user, single-keyword search, only supports one interaction, and does not sort the search results. Under this background, the user efficiency is not high, and the searchable encryption technology faces engineering applications and promotion. To solve this problem, we propose a searchable encryption scheme that supports grouping and sorting. The keywords corresponding to the file are uploaded to the cloud server in order, and the upload order of each keyword in the file is recorded. Stored in the index structure, the index is stored in the form of keyword-keyword sequence number-file address pointer. When the user searches, the keywords are first matched, and then the order of the keywords in the matching file is determined. The files with the same keyword sequence number should be grouped and sorted to realize the function of grouping and sorting.

The main advantages of the solution in this article:

1. Design a multi-keyword searchable encryption scheme to realize multi-user search, and add the information of the keyword sequence while establishing the keyword index.
2. Sort the searchable encrypted search results, which can sort the results according to the degree of relevance, and further improve the search efficiency.
3. The index is added, deleted and other technologies, so that the searchable encryption technology has better practicability.
4. Through the bilinear hypothesis for difficult problems, the safety of the search process is proved.
5. Through the comparison of simulation calculations, searchable encryption using the adjacent index structure has certain advantages in terms of storage and computational overhead.

2 Related Research

Searchable encryption technology was first proposed by Song et al. [1]. This scheme opened the research process of searchable encryption, but this scheme could not achieve provable security. Under the condition of strong provable security, in 2006, Curtmola et al. [2] proposed the current searchable symmetric encryption with the best performance, but the length of the retrieval trapdoor and the maximum number of documents in this scheme are linear; literature [3–6] realized multi-keyword column query and multi-dimensional range query. Lu et al. [7] improved the search efficiency of range retrieval by constructing an index structure. In recent years, in recent years, dynamic SSE research has received extensive attention. In 2012, Kamara et al. [8] first proposed a dynamic

searchable symmetric encryption (DSSE, dynamic searchable symmetric encryption) solution that supports dynamic updates, and formalized the security of DSSE. DSSE not only supports ciphertext retrieval, but also supports dynamic update operations of searchable ciphertext, such as ciphertext addition, ciphertext deletion, and keyword addition and deletion.

Wang et al. [4] solved the problem of sorting and searching encrypted data for the first time, returning matching files through search and sorting, which greatly enhanced the usability of the system. Fu et al. [9] first studied and solved the problem of personalized multi-keyword sorting search for encrypted data. Zhang et al. [10] proposed a multi-keyword sorting search scheme in the multi-owner model. Wang et al. [11] proposed a verifiable fuzzy keyword search scheme based on symbol tree, which has the verifiability of search results. In 2013, Yu et al. [12] proposed a dynamic sorting SE scheme, which stores the tf value and idf value of each file in the inverted index. When updated, only the idf value is used, and the updated key will be recalculated. Therefore, instead of updating all file vectors, only the auxiliary vector is updated. In 2015, Orencik and Savas [13] proposed a multi-keyword ranking, but it could not provide an accurate ranking. In the same year, Zhang et al. [10] proposed a searchable encryption technology to protect keyword ranking information. This solution achieved the ranking of search results while also protecting the ranking information on the cloud server. In 2017, Miao et al. [14]. The searchable encryption technology is realized through the method based on the attribute, and the sorting of the document level is realized. In 2020, Peng Haiyang and others [15] used segmented index to record the control and management of the search range in the index structure in a contiguous manner, but did not sort the search results. In 2020, Guan et al. [16] designed a The cross-language multi-keyword sorting search with semantic expansion can also speed up the sorting process by designing top-k data. The retrieval protocol is based on a heap binary tree structure.

3 Mathematical Foundation

3.1 Bilinear Pairs

Suppose that $\mathbf{G_1}$ and $\mathbf{G_2}$ are both multiplicative cyclic groups of prime number ρ, and g is the generator of G_1. We call the mapping: $\mathbf{G_1} \times \mathbf{G_2} \rightarrow \mathbf{G_2}$ is a bilinear mapping. If the mapping $\mathcal{C}$satisfies the following 3 properties.

(1) Bilinear. For any $a, b \in Z\rho$, $g \in \mathbf{G}$, there is $(g^a, g^b) = (g, g)^{ab}$. For any $g_1, g_2, g \in \mathbf{G}$, there is $(g_1, g_2, g) = (g_1, g)\,(g_2, g)$. For any $g_1, g_2, g \in \mathbf{G}$, there is $(g, g_1 g_2) = (g, g_1)\,(g, g_2)$.
(2) Non-degeneration. $(g, g) \neq 1$.
(3) Computable. For any $\mathbf{P}, \mathbf{Q} \in \mathbf{G}$, it can be calculated that $(\mathbf{P}, \mathbf{Q})$ belongs to $\mathbf{G_2}$.

3.2 Introduction to Adjacency Index

In order to sort the search results in searchable encryption, the order-preserving encryption method can usually be used to sort the order of the search results. The order-preserving encryption is not flexible enough to deal with complex scenarios; it can also appear in the file by keyword. When performing the search process, the results of the documents are sorted according to the frequency of keywords. However, this solution first needs to calculate the frequency of keyword occurrences. The frequency of keyword occurrences does not occur in some scenarios. It does not represent the degree of relevance, and some commonly used words will also occupy a large frequency in the text, and the keyword frequency requires full-text scanning, which brings a certain degree of complexity to the calculation.

This article uses adjacency index. In view of the characteristics of the adjacency index that can store other description information on the index, the secret keywords are linked by a linked list, and the documents with the keywords in the same position are grouped into a group and uploaded according to the sequence number of the keywords. Perform grouping and sorting. In this way, the search results can be sorted relatively quickly, and a group of files with a higher degree of relevance can be found quickly, which improves search efficiency. At the same time, this solution is also applicable to multi-keyword search scenarios.

3.3 Construction, Update, and Deletion of Adjacency Index

The data owner (Do) first identifies the document and establishes a set, which can be expressed as: $\mathbf{D} = \{d_1, d_2, \ldots, d_n\}$, and extracts keywords from the file. The set of keywords can be expressed as $\mathbf{W} = \{w_1, w_2, w_3, \ldots, w_n\}$, form the corresponding index, where the file d_x contains multiple keywords and arranged in order $\{w_{x1}, w_{x2}, \ldots\ldots w_{xn}\}$, w_{x1} means that the keyword w_x is in this The first keyword uploaded in the file is used to infer and upload all the keywords contained in the file..

As shown in Fig. 1 Adjacency index structure diagram, the keywords of all documents are combined and encrypted to form an index. The left side of w_1 links to the addresses of all documents containing the keyword w_1, and w_{11} is adjacent to all documents that contain the keyword w_1 and the keyword w_1 is the first upload among all documents The set $\{d_{111}, d_{112}, \ldots, d_{11n}\}$ of all file set identifiers, as shown in Fig. 1, for example; w_{12} represents the file containing the keyword w_1 and w_1 is the second uploaded keyword, that is, the keyword In the process of document uploading, w_1 is the second uploaded keyword. The set of identifiers for all files is $\{d_{111}, d_{112}, \ldots, d_{11n}\}$, and so on, the meaning contained in the adjacent index structure in the figure can be obtained.

Extract all keywords and use contiguous indexing, and add a keyword sequence identifier (w_{xy}, where w_x represents the keyword, and y represents the upload order of the keyword in the file, that is, keyword Serial number), each keyword sequence identifier links the file ID collection that contains the keyword and the keyword is in the same keyword sequence number in the file, which constitutes an important part of this article. The adjacent index is composed of when performing a search, the keywords are first matched, and then the corresponding files are grouped according to the relevant sequence number of the keyword to obtain the corresponding files. After the files are

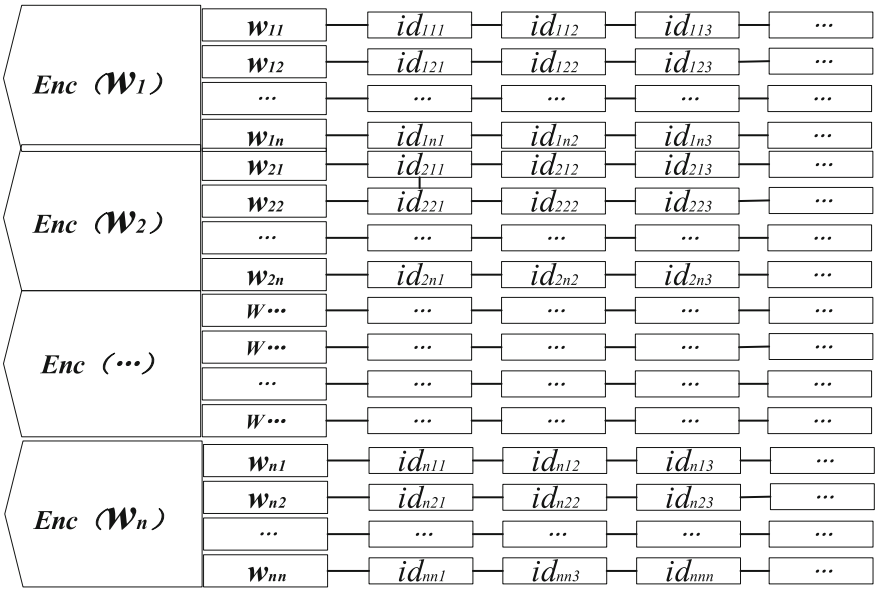

Fig. 1. Adjacency index structure diagram

obtained and they are small to large according to the keyword sequence number, all files are in the form of keyword sequence number-document Perform grouping and sorting.

Update, first verify the legal identity of the user, collect the document keywords and sort the keywords, find the position of the keyword in the index according to the keyword information, and determine the sequence number of the keyword in the file, and add it to Correspond to the file collection under the serial number (usually added to the end of the linked list), As shown in Fig. 2 Adjacency index update graph, and map the corresponding file address with the keyword serial number (as shown in the Fig. 2). By orderly updating the file ids on the index, it is possible to ensure real-time and effective ordering of the updated files.

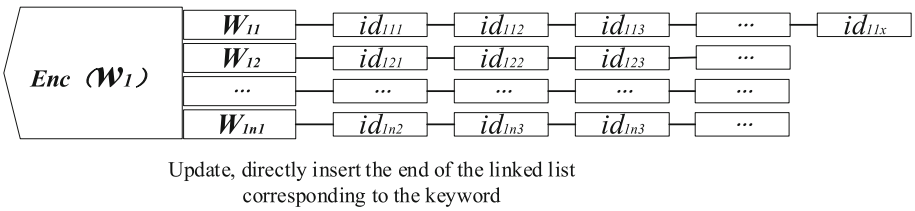

Fig. 2. Adjacency index update graph

To delete the index, first verify the legal identity of the user, then find the keyword through the index, As shown in Fig. 3 Adjacency index deletion graph, find the position of the file in the index according to the keyword sequence number in the file, and point

the pointer of the file to the next file. If you want to delete The file of is at the end of the index, just delete it directly.

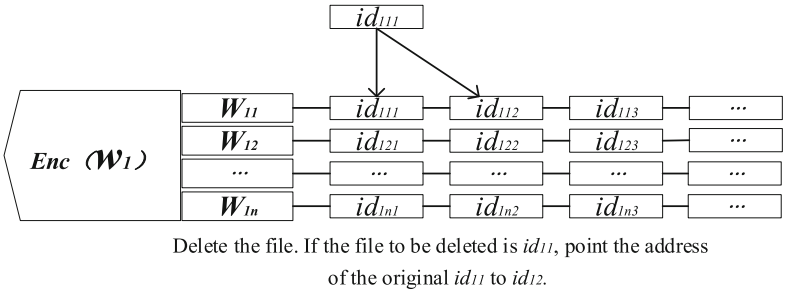

Fig. 3. Adjacency index deletion graph

4 System Model

4.1 Introduction to System Basic Model

The solution model is composed of data owners, cloud servers, group users, and internal organizations. Through the cooperation of these parts, the search results can be sorted by groups, according to the order of keywords in the document, and according to the order of keywords in the document. The order in a single document is collected and then sorted.

1. Data owner: In this system, it is a user group with multiple sub-users. The data owner can also be a data user. The data user uploads the keywords contained in the file in an orderly manner and uploads in order of relevance. It generates an index on the cloud server, and then the encrypted file is uploaded to the cloud server.
2. Cloud server: After the user logs in to the server with the user name and password, the cloud server stores the encrypted files uploaded by the user, stores the encrypted index, performs searchable and encrypted search process, and returns the search results to the user who submitted the search request.
3. User group: The user group is the user of data. The data user remotely logs in to the server through his own private key, keywords, user name and password assigned by the internal organization and submits a search request to the server, and the server returns the search results to the user.
4. Internal organization: The internal organization is credible, generates public parameters and assigns user names and passwords to users, and maintains the authority for cloud server and user authentication (Fig. 4).

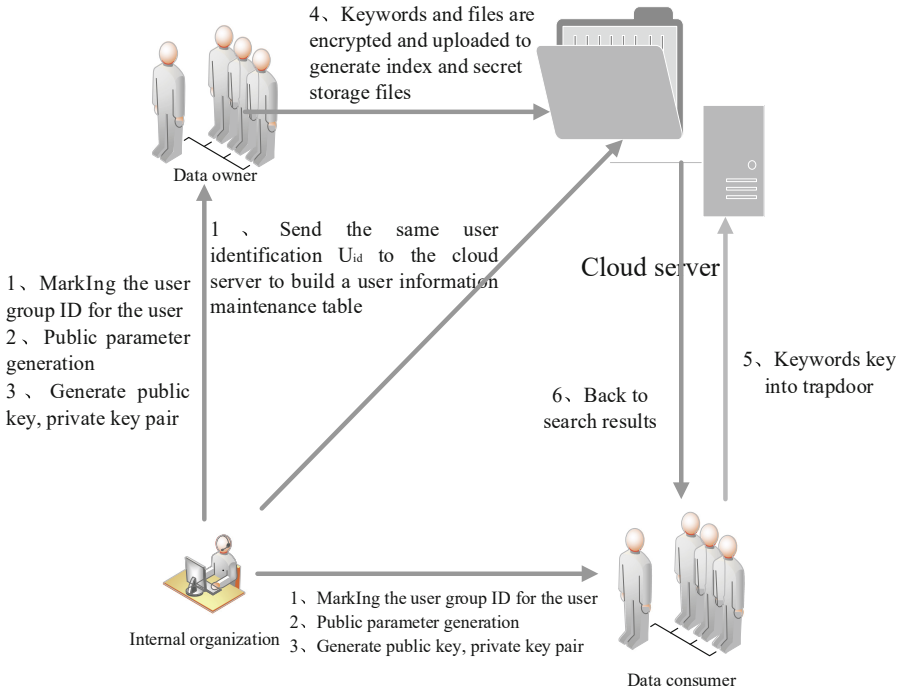

Fig. 4. Searchable encryption scheme composition diagram

4.2 Introduction to the Formal Process

The design purpose of this article is to design a searchable encryption scheme that can group and sort search results according to the location information of keywords, by sorting the file keywords before uploading, and then recording the order information of the keywords in the file to the index In the structure, when the user requests a search, the keyword is generated trapdoor, and the cloud server executes the search process, and matches the file corresponding to the keyword, and then sorts the keywords by the order of the keywords in the document, and finally realizes the search The order of the results.

1) $ParaGen(1^n) \rightarrow \mathbf{P}$, Mpk, Msk. The algorithm is executed by an internal organization. A security parameter n is input, and the public parameter P of the system is output, the master public key Mpk, and the master private key Msk.
2) $Accredit(\boldsymbol{U}) \rightarrow U_{id}$, The algorithm is run by an internal organization, assigns different identity information to each legal user, and internally organizes and maintains user information used by the system.
3) $KeyGen(\mathbf{P}, \text{Mpk}, \text{Msk}, \text{U}_{\text{id}}) \rightarrow pk_i, sk_i$. The algorithm inputs the public parameters **P**, the master key Mpk, the master private key Msk, the unique identifier Uid assigned by the internal organization, and outputs the legal user's public key and private key pair (pk_i, sk_i).
4) $Build\ Index(\mathbf{P}, \mathbf{W}, pk_i, U_{id}) \rightarrow$ Index (**W**) index establishment, enter the public parameter **P** generated by the system, the public key pk_i of the data owner, and

the keyword information **W** of the file, where **W** contains multiple keys Keyword information group with a certain sequence, unified identity $U_{id,}$ first check whether the unified identity U_{id} of the data owner is in the user information table, if not, refuse service upload; if yes, the keyword is encrypted to construct an adjacency index, The output index Index (**W**).

5) *Enc*(pk_i, M) → E(c) Enter the uploader's public key and file for encryption (it can also be a symmetric key), and enter the cipher text m = E(c).
6) *Trapdoor*(U_{id}, w_x, sk_i) → T(w_x), enter the keyword w_x requested by the user and the user's private key sk_i to generate a keyword trapdoor T(w_x).
7) *Test*(T(sk_i), *Index*, U_{id}). Enter the trapdoor and the index *Index* (***W***), the unique identifier U_{id}, and perform the following process: Query the user in the user information table on the server. If the user exists in the information table, the index ciphertext match will be performed to find the keyword Existing ciphertext, and sort the search results by the sequence structure of the keywords, obtain the accessible file addresses, and find the corresponding ciphered files through the keyword and file mapping relationship.

5 Security Proof

5.1 Bilinear Difficult Problem

A variant of the bilinear Diffie-Hellman problem, Given two multiplications of order of large prime q The cyclic group is a generator of, and the bilinear mapping problem is: for a given where calculation If the adversary of any polynomial time solves the problem with a negligible advantage, we consider the problem to be difficult, which is formalized as:

$$\Pr\left[A\left(g, g^a, g^b, g^c, g^{1/a}\right) = e(g,g)^{abc}\right] \geq \varepsilon \tag{1}$$

5.2 Security Certification Process

The security target attacker selects any user U_{id} within the range and the trapdoor information *T(w)* corresponding to the keyword w that the user can obtain. The main idea is to prove that the attacker cannot obtain effective information about the keyword corresponding to the trapdoor through the known legal trapdoor $T(w)$. In the simplest way, it can be considered that the attacker cannot obtain the effective information from the two legal trapdoors. In the trap door, it is effective to judge whether the trap door comes from the same keyword [2].

System initialization: The system first runs the algorithm to generate the public parameters required by the system, including the master key sk_0 and the master private key pk_0, assigns the corresponding user public and private key pair sk_{ui}, pk_{ui} to the data user, and generates the public and private key pair.

Inquiry and challenge phase 1:

First inquiry: The attacker randomly selects a keyword w to generate a trapdoor to initiate an inquiry. The cloud server calculates the trapdoor $T(w)$ of the keyword

w through the trapdoor generation algorithm, and returns the calculated result to the attacker.

Inquiry and Challenge Phase 2:

The attacker again initiates an inquiry and challenge to the ciphertext and index, but the keywords that initiate the inquiry and challenge cannot be the keywords that appeared in the inquiry and challenge phase 1, and the calculated trapdoor is returned to the attacker.

Inquiry phase 3:

The attacker randomly selects one of the two keywords and initiates a query. The query keyword cannot be a keyword that has already been queried. The simple expression is: w_b, $b \in \{0, 1\}$, Use the attacker's key to encrypt and send it to the cloud server.

Output: The attacker outputs guesses $b' \in \{0, 1\}$, If the attacker's guess $b' \in b$ is correct, the attacker is at an advantage during the entire game challenge, and the probability of winning the game is:

$$\varepsilon = \left| \Pr(b' = b) - \frac{1}{2} \right| \tag{2}$$

The hypothesis based on the bilinear difficult problem is: if the advantage of any polynomial time adversary winning in the game is that ε is a negligible function, then the search trapdoor of this scheme meets the trapdoor under the adaptive keyword attack. Distinguishability. That is to say, for the given two trapdoors, the attacker cannot guess the trapdoor corresponding to the keyword from the known information with a non-negligible advantage, and it is also safe to resist semantic attacks.

6 Plan Analysis and Comparison

This paper selects the searchable encryption field in the past few years and the similar literature [10, 14] for comparison, in the case of achieving similar results, from the comparison of computing overhead and storage overhead, analyze the comparison between this paper and theirs. Advantage.

For more convenience, use t_{eo} and t_{e1} to represent the computational cost of performing an exponential operation in G_0 and G_1, respectively, use t_{Ho} and t_{H1} to represent the computational cost of performing a hash operation that maps an arbitrary string to G_0 and G_1. In addition, $|G_0|$ and $|G_1|$ respectively represent the length of the elements of the group G_0 and G_1, and |Z| represents the length of the ciphertext of the symmetric encryption algorithm. Through the unified definition of parameters, this article can intuitively compare the various solutions.

The comparison results are shown in Table 1. Through analysis, it is easier to see that the trapdoor size of Zhang's [10] scheme is $(1 + u)|\boldsymbol{G}_0|$, while Miao's scheme [14] and this scheme are $(2y + 4)|\boldsymbol{G}_0|$. In Zhang's [10] scheme, the storage cost it grows as the number of keywords in the query increases, and this solution increases with the number of keywords. For example, when a large number of keywords are queried and only a small amount of grouping and sorting is involved, the storage overhead of this solution has certain advantages in terms of storage. In addition, the ciphertext size in the Miao [14] scheme is $|F||G_0| + (|F| + jN + N)|G_1|$. The ciphertext size of this scheme

Table 1. Comparing computational overhead and storage overhead

	Miao's plan	Zhang's plan	This plan
Index storage	$(2x+2+n)\|G_0\|$	$2n\|G_0\|$	$(2x+n)\|G_0\|$
Trapdoor storage	$(2y+4)\|G_0\|$	$(1+u)\|G_0\|$	$(2y+4)\|G_0\|$
Ciphertext storage	$\|F\|\|G_0\|+(\|F\|+jN+N)\|G_1\|$	$2n\|G_{\mathbf{0}}\|$	$k\|G_{\mathbf{0}}\|+(k+N)\|G_{\mathbf{1}}\|$
Keygen computing	$(2y+3)t_{e_0}+yt_{H_1}$	nt_{H_0}	$(2y+3)t_{e_0}+yt_{H_1}$
Enc computing	$(2\|F\|+2x+n)t_{e_0}+xt_{H_1}$	$2nt_{e_0}$	$(2\|F\|+n)t_{e_0}+xt_{H_1}$
Trapdoor computing	$(2y+4)t_{e_0}$	$(1+u)t_{e_0}$	$(2y+2)t_{e_0}$
Decryption computing	$\|F\|t_{e_0}$	–	$\|F\|t_{e_0}$

Note: *x: the number of attributes in the system; Y: the number of data user attributes; |F|: the number of undifferentiated files; N: the number of transmission nodes: k: the number of top-k files; j: the number of child nodes of the transmission node; n: the number of keywords in the keyword dictionary; U: the number of keywords queried by the user*

is $k|G_0|+(k+N)|G_1|$. In a massive data system, the value of k is obviously much smaller than $|F|$. In this way, compared with the Miao scheme [14], this scheme also has advantages in terms of ciphertext size.

In the table, for the calculation cost of the KeyGen, Enc, Enc Tpdr algorithm, this scheme requires $(2y+3)t_{eo}+yt_{H1}$, $(2|F|+n)t_{eo}+xt_{H1}$, $(2y+2)t_{eo}$, From these data comparisons. It is hard to see that compared with the computational cost of the Miao scheme [14], in a large-scale data sharing system, n is the number of files, much larger than other values, the algorithm in this scheme has higher computational efficiency, which means that this scheme is more effective and practical.

7 Summary and Outlook

Aiming at the search scenario of a large number of users and files, this paper proposes a searchable encryption scheme based on keyword location grouping and sorting. Through the adjacent index structure, the order information of the keywords used and the file are mapped and associated with each other, so that the search results are grouped and sorted In the scheme, the index structure of this scheme uses adjacency, which is more efficient and easy to maintain. In terms of safety, this paper adopts the assumption of difficult problems based on bilinear pairs, and the safety has been proved. Through literature surveys, although the solution proposed in this article has certain advantages in some scenarios, it has certain advantages in computing overhead and storage overhead through comparison, but it has not been further optimized in a single group. This article is more critical The scheme of word ordering in this article is also applicable, but it is not explained and introduced in detail. These application scenarios are also urgently needed schemes in reality. In the future, we will do further research on these problems.

Acknowledgements. This work was supported by Sichuan Science and Technology Program (2020YFG0298), Sichuan Science and Technology Program (2021JDRC0077), and Key Laboratory Fund (6142103010711).

References

1. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy, S&P 2000, pp. 44–55. IEEE (2000)
2. Curtmola, R., Garay, J., Kamara, S., et al.: Searchable symmetric encryption: improved definitions and efficient constructions. J. Comput. Secur. **19**(5), 895–934 (2011)
3. Nepolean, D., Karthik, I., Preethi, M., Goyal, R., Vanethi, M.K.: Privacy preserving ranked keyword search over encrypted cloud data. In: Martínez Pérez, G., Thampi, S.M., Ko, R., Shu, L. (eds.) SNDS 2014. CCIS, vol. 420, pp. 396–403. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54525-2_35
4. Wang, C., Cao, N., Li, J., et al.: Secure ranked keyword search over encrypted cloud data. In: 2010 IEEE 30th International Conference on Distributed Computing Systems, pp. 253–262. IEEE (2010)
5. Wang, C., Cao, N., Ren, K., et al.: Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans. Parallel Distrib. Syst. **23**(8), 1467–1479 (2011)
6. Cao, N., Wang, C., Li, M., et al.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**(1), 222–233 (2013)
7. Lu, Y.: Privacy-preserving logarithmic-time search on encrypted data in cloud. In: NDSS (2012)
8. Kamara, S., Papamanthou, C., Roeder, T.: Dynamic searchable symmetric encryption. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 965–976 (2012)
9. Fu, Z., Ren, K., Shu, J., et al.: Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans. Parallel Distrib. Syst. **27**(9), 2546–2559 (2015)
10. Zhang, W., Lin, Y., Xiao, S., et al.: Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing. IEEE Trans. Comput. **65**(5), 1566–1577 (2015)
11. Wang, J., Chen, X., Ma, H., et al.: A verifiable fuzzy keyword search scheme over encrypted data. J. Internet Serv. Inf. Secur. **2**(1/2), 49–58 (2012)
12. Yu, J., Lu, P., Zhu, Y., et al.: Toward secure multi keyword top-k retrieval over encrypted cloud data. IEEE Trans. Dependable Secure Comput. **10**(4), 239–250 (2013)
13. Orencik, C., Alewiwi, M., Savas, E.: Secure sketch search for document similarity. In: 2015 IEEE Trustcom/BigDataSE/ISPA, no. 1, pp. 1102–1107. IEEE (2015)
14. Miao, Y., Ma, J., Liu, X., et al.: Attribute-based keyword search over hierarchical data in cloud computing. IEEE Trans. Serv. Comput. **13**(6), 985–998 (2017)
15. Peng, H., Dong, G., Yao, H., Zhao, Y., Chen, Y.: Searchable encryption scheme with limited search scope for group users. In: 2020 International Conference on Networking and Network Applications (NaNA), Haikou City, China, pp. 430–435 (2020). https://doi.org/10.1109/NaNA51271.2020.00079
16. Guan, Z., Liu, X., Wu, L., et al.: Cross-lingual multi-keyword rank search with semantic extension over encrypted data. Inf. Sci. **514**, 523–540 (2020)