Privacy-Preserving Multi-keyword Search over Outsourced Data for Resource-Constrained Devices

Lin-Gang Liu[1], Meng Zhao[2], Yong Ding[1,5(✉)], Yujue Wang[1], Hua Deng[3], and Huiyong Wang[4]

[1] Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
stone_dingy@126.com
[2] School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, China
[3] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[4] School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China
[5] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China

Abstract. With the rapid development of cloud computing, a variety of cloud-based applications have been developed. Since cloud computing has the features of high capacity and flexible computing, more and more users are motivated to outsource their data to the cloud server for economic savings. Users are able to search over outsourced data according to some keywords with the help of the cloud server. During data searching, the confidentiality of the relevant data could be compromised since the keywords may contain some sensitive information. However, existing privacy-preserving keyword search proposals have high computation complexity, which are not applicable to IoT-related scenarios. That is, the data processing and search trapdoor generation procedures require the users to take resource-intensive computations, e.g., high-dimensional matrix operations, which are unaffordable by resource-constrained devices. To address this issue, we propose a light-weight privacy-preserving multi-keyword search scheme. The security and performance analyses demonstrate that our scheme outperforms existing solutions and is practical in applications.

Keywords: Cloud computing · Outsourced data · Keywords search · Data privacy · Internet of Things

1 Introduction

With the advent of cloud computing, the users with limited local resources do not need to purchase expensive hardware to support massive data storage. Thus,

Z. Zheng et al. (Eds.): BlockSys 2020, CCIS 1267, pp. 282–294, 2020.
https://doi.org/10.1007/978-981-15-9213-3_22

for economic savings, more and more individuals and enterprises engage cloud servers to maintain their data. However, users would lose control of outsourced data, which may leak some sensitive information, for example, in the cases where health records and private emails are hosted on cloud servers. Therefore, to protect data privacy, user data should be stored on the cloud server in ciphertext format.

For retrieving the interested data, users can request the cloud server to search over outsourced dataset with some specific keywords. However, the keywords may contain some sensitive information of outsourced data, which means these keywords cannot be presented to the cloud server in plaintext format, otherwise the users' private information could be deduced by the cloud server. To address this issue, privacy-preserving keyword search has recently gained attention, and many solutions have been proposed [8,10,17,26].

However, existing solutions require users to take heavy computations in both phases of data processing and search trapdoor generation. For example, in [4], users need to perform high-dimensional matrix operations, such as multiplications and inversion, where the matrix dimension is determined by the cardinality of the keyword set. While in [20], users have to compute many exponentiation operations for generating searchable indexes and search trapdoor. To deploy in the Internet of Things setting, existing works are not suitable since those heavy computation operations are not affordable by resource-constrained devices.

1.1 Our Contributions

To address the above issue, this paper proposes a light-weight scheme supporting privacy-preserving ranked multi-keyword search over outsourced data, where the search results are determined by the similarity score between the search query and the keyword index of outsourced data. Our contributions are summarized as follows.

- The proposed scheme allows the user to search for outsourced data with multiple keywords, and the search results can be ranked so that the cloud server only needs to return the results satisfying the given threshold.
- The proposed scheme can guarantee the privacy of searchable index of outsourced data and queries. That is, the cloud server cannot deduce any private information of outsourced data from the encrypted index and queries.
- The proposed scheme can guarantee the unlinkability of search trapdoors. That is, for two search trapdoors submitted by the user, the cloud server is unable to identify whether they are generated for the same query.
- In both data processing and query generation phases, the user only needs to take light-weight computation operations.

Performance analysis demonstrates that our scheme is much more efficient than existing solutions, thus it can be deployed in IoT setting to support resource-constrained devices.

1.2 Related Works

The first single keywords searchable encryption scheme over outsourced encrypted data was proposed by Song et al. [21] in the symmetric key setting. Subsequently, a lot of this type schemes [2,6,15,23] were designed. David et al. [5] proposed a scheme supporting single-keyword boolean search over large outsourced dataset. However, the single keyword search mechanism cannot provide accurate search results. Since the cloud server usually stores massive data, there would be many match data satisfying the search condition of a single keyword, and most of the search results may have no relation with the expected data.

To support more sophisticated outsourcing search methods, many multi-keyword search schemes have been proposed [4,9,11,14]. These schemes can allow the cloud server to return the most relevant data, thus, they are more practical than the single keyword search mechanism in supporting real-world applications. Multi-keyword search can allow complicated search conditions on outsourced data, for example, the works [1,16] support multi-keyword search with fully homomorphic encryption, [11,14] support conjunctive keyword search, [9] supports multi-keyword fuzzy search, and [4] supports ranked multi-keyword search.

In the public-key setting, the first searchable encryption scheme was proposed by Boneh et al. [3], where anyone can outsource encrypted data to the cloud server, but only the user holding the private key can issue search queries. Xu et al. [25] constructed a searchable public-key ciphertexts scheme with hidden structures to achieve fast search. Hu et al. [13] presented a public-key encryption scheme with keyword search from obfuscation, where the cloud server is given an obfuscated simple decrypt-then-compare circuit with the secret key to perform keyword search. Xu et al. [24] designed a public-key multi-keyword searchable encryption scheme with a hidden structures model, which also supports boolean search over encrypted e-mails. Wang et al. [22] proposed a tree-based public-key multi-dimensional range searchable encryption scheme from the predicate encryption method and leakage function.

To enrich the functionality of searching over remote data, various practical schemes have been designed. He and Ma [12] proposed a fuzzy search scheme over encrypted data using bloom filter. Zhang et al. [27] noticed that He and Ma's proposal [12] cannot resist the sparse non-negative matrix factorization based attacks, and further presented a multi-keyword fuzzy search scheme using random redundancy method. Fu et al. [10] designed a semantic-aware search scheme, where both the index and search trapdoor contain two vectors.

Cao et al. [4] proposed an efficient multi-keyword ranked search scheme over encrypted cloud data, where coordinate matching was introduced to capture the relevance between data documents and the search query. In Raghavendra et al.'s solution [19], the index for keywords was generated using split factor, and to save computation overheads, the index tree was constructed to store keywords. Ren et al. [20] studied multi-keyword ranked search, where the search trapdoor is generated using a polynomial function. Ding et al. [7] constructed a keyword set using k-grams and Jaccard coefficient, and also built searchable index of small

size. In Liu et al.'s scheme [17], the user is allowed to update the outsourced data and verify the search result.

1.3 Paper Organization

The remainder of this paper is organized as follows. Section 2 describes our system model, threat model, and design goals. Section 3 presents our scheme, which security and performance are evaluated and compared in Sect. 4. Section 5 concludes the paper.

2 Problem Formulation and Design Goals

2.1 System Model

As shown in Fig. 1, a multi-keyword search system consists of three types of entities, that is, data owner, data user and cloud server. There is a secure communication channel between data owner and data user. The data owner outsources a collection of documents to the cloud server. Since the documents may contain sensitive information, they cannot be directly uploaded to the cloud server. Thus, to protect the privacy of outsourced documents, they should be outsourced in ciphertext format.

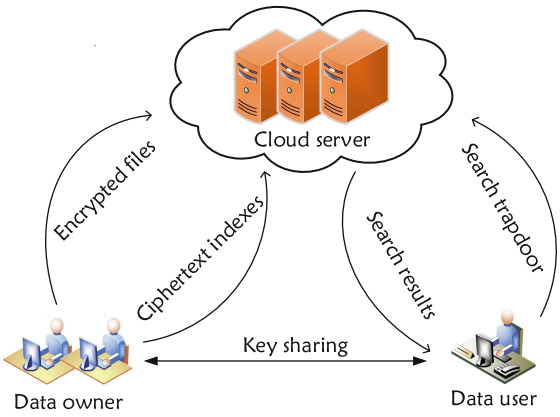

Fig. 1. The system model

To facilitate data searching, the outsourced documents should be attached with a list of keywords. All keywords are contained in a keyword dictionary. To guarantee that the keywords cannot leak the privacy of outsourced documents, in the data processing phase, the data owner is able to produce an encrypted searchable index for each document. The searchable index is outsourced to the cloud server along with the document.

In the search phase, data user can generate a search trapdoor of its query vector with multiple keywords to enable the cloud server to search over outsourced documents. The keywords in the query vector are also contained in the keyword dictionary, which should be transformed into search trapdoor to protect the privacy of outsourced data. Upon receiving the search trapdoor, the cloud server computes the similarity score between each encrypted searchable index and the search trapdoor, and returns the document if its similarity score satisfies the given search threshold.

2.2 Threat Model

In the honest-but-curious model, the cloud server can perform multi-keyword search according to the user's request, but it is curious about the sensitive information of outsourced documents. That is, the cloud server may try to deduce some information from the outsourced documents, ciphertext indexes, and search trapdoors. This paper assumes the adversary is able to launch known ciphertext attacks and known background attacks on outsourced documents.

Known ciphertext attack: The cloud server only knows some ciphertext information including encrypted documents, ciphertext indexes and search trapdoors. With these information, the cloud server aims to get the sensitive information of outsourced documents.

Known background attack: The cloud server may also know more background information of outsourced documents, such as statistic information of documents and relation of search trapdoors. These background information may leak the search pattern to the cloud server.

2.3 Design Goals

A secure multi-keyword ranked search scheme needs to satisfy the following requirements.

- *Data privacy*: The cloud server should not be able to infer any information about outsourced documents.
- *Keyword privacy*: The cloud server should not determine whether a specific keyword is relevant to a outsourced document according to encrypted document, encrypted index, search trapdoor and background knowledge.
- *Trapdoor unlinkability*: The cloud server should not be able to identify whether two search trapdoors are generated from the same query.
- *Efficiency*: Due to the limited computation capability of data owner and data user, the data processing and query generation phases cannot contain resource-intensive computations.

3 Concrete Construction

This section introduces a light-weight and privacy-preserving multi-keyword search scheme based on the inner product similarity computing scheme [18]. Some notations and the corresponding descriptions are given in Table 1.

Table 1. Notations and descriptions

Notations	Descriptions
F	Document set $F = \{F_1, F_2, ..., F_m\}$
$\overline{F}$	Encrypted document set $\overline{F} = \{\overline{F}_1, \overline{F}_2, ..., \overline{F}_m\}$.
W	Keyword dictionary
$\overline{W}$	A set of search keywords
$\overline{I}$	A plaintext index vector $\overline{I} = (\overline{I}_1, \overline{I}_2, ..., \overline{I}_m)$
$\hat{I}$	A ciphertext index vector $\hat{I} = (\hat{I}_1, \hat{I}_2, ..., \hat{I}_m)$
$Q_{\overline{W}}$	Query vector $Q_{\overline{W}} = (Q_{\overline{W}_1, \overline{W}_2, ..., \overline{W}_n})$ constructed from $\overline{W}$
$\hat{Q}_{\overline{W}}$	Search trapdoor in ciphertext format
S	The secret key of data owner
N_i	The filename of document F_i
d_i	The file size document F_i
γ_i	The hash value with regard to document F_i

- **System setup**: With input security parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, the data owner constructs a dictionary W, which contains n keywords. The data owner randomly picks a large prime numbers p such that $|p| = \lambda_2$, $S \in_R Z_p^*$, and a cryptographic one-way hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_1}$. Thus, the public parameters are $para = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, p, n, H)$, and the data owner keeps W and S secret.
- **Index generation**: For each document F_i $(i = 1, ..., m)$, the data owner encrypts it as ciphertext document $\overline{F}_i$ using some secure symmetric encryption, randomly picks a unique file name N_i, and calculates the length d_i of document F_i. The data owner computes $\gamma_i = H(N_i, d_i)$ and constructs the index vector $\overline{I}_i$ such that if the document F_i contains the jth keyword in the dictionary W, then $\overline{I}_{i,j} = 1$, otherwise $\overline{I}_{i,j} = 0$. The data owner further sets $\overline{I}_{i,n+1} = 0$ and $\overline{I}_{i,n+2} = 0$, chooses $n+2$ random number m_j such that $|m_j| = \lambda_3$ for $1 \leq j \leq n+2$, and encrypts each $\overline{I}_{i,j}$ as follow:

$$\hat{I}_{i,j} = S \cdot (\overline{I}_{i,j} \cdot \gamma_i + m_j) \mod p \tag{1}$$

  Then for document F_i, the data owner outsources the ciphertext index vector $\hat{I}_i = (\hat{I}_{i,1}, \hat{I}_{i,2}, \cdots, \hat{I}_{i,n+2})$ and the processed file $\hat{F}_i = (\overline{F}_i, \gamma_i)$ to the cloud server, and keeps (N_i, d_i) at local.
- **Trapdoor generation**: Data user picks a large random number δ such that $|\delta| = \lambda_1$, and computes $S^{-1} \mod p$. From the query keyword set $\overline{W}$, data user constructs query vector $Q_{\overline{W}}$, where $Q_{\overline{W}_j} = 1$ if the query keyword set $\overline{W}$ contains the jth keyword in the dictionary W, otherwise $Q_{\overline{W}_j} = 0$. Data user then sets $Q_{\overline{W}_{n+1}} = 0$ and $Q_{\overline{W}_{n+2}} = 0$, and randomly chooses $n+2$ numbers t_j such that $|t_j| = \lambda_4$ for $1 \leq j \leq n+2$. Data user constructs the search trapdoor $\hat{Q}_{\overline{W}}$ as follows.

$$\hat{Q}_{\overline{W}_j} = S^{-1} \cdot (Q_{\overline{W}_j} \cdot \delta + t_j) \mod p \tag{2}$$

Data user sets search threshold τ, and submits the search trapdoor $\hat{Q}_{\overline{W}} = (\hat{Q}_{\overline{W}_1}, \hat{Q}_{\overline{W}_2}, \cdots, \hat{Q}_{\overline{W}_{n+2}})$ and (τ, δ) to the cloud server.

– **Search**: Once received the encrypted search trapdoor $\hat{Q}_{\overline{W}}$, the cloud server computes the similarity score $Score(\overline{I}_i, Q_{\overline{W}})$ with each $\hat{I}_i$ as follows. The cloud server computes

$$E_i = Score(\hat{I}_i, \hat{Q}_{\overline{W}}) = \hat{I}_i \cdot \hat{Q}_{\overline{W}} \mod p \tag{3}$$

By properly choosing the elements under the given security parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, we assume both the following conditions hold

$$\hat{I}_i \cdot \hat{Q}_{\overline{W}} < p$$

and

$$\begin{aligned}\rho = &\sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}\neq 0}^{n+2} (\gamma_i t_j \overline{I}_{i,j} + m_j \delta Q_{\overline{W}_j} + m_j t_j) + \\ &\sum_{j=1, I_{i,j}=0, Q_{\overline{W}_j}\neq 0}^{n+2} (m_j \delta Q_{\overline{W}_j} + m_j t_j) + \sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}=0}^{n+2} (\gamma_i t_j \overline{I}_{i,j} + m_j t_j) + \\ &\sum_{j=1, I_{i,j}=0, Q_{\overline{W}_j}=0}^{n+2} m_j t_j \\ < &\gamma_i \delta.\end{aligned}$$

Then, the cloud server computes

$$Score(\overline{I}_i, Q_{\overline{W}}) = \sum_{j=1}^{n} \overline{I}_{i,j} \cdot Q_{\overline{W}_j} = \frac{E_i - (E_i \bmod \delta \cdot \gamma_i)}{\delta \cdot \gamma_i} \tag{4}$$

If the following search condition is satisfied

$$Score(\overline{I}_i, Q_{\overline{W}}) \geq \tau$$

then the cloud server returns the corresponding document $\overline{F}_i$.

Theorem 1. *The proposed multi-keyword search scheme is correct.*

Proof. To compute the similarity score $Score(\overline{I}_i, Q_{\overline{W}})$, it is required that both $\overline{I}_{i,j} \neq 0$ and $Q_{\overline{W}_j} \neq 0$ are satisfied for $1 \leq j \leq n$. Let

$$E_i' = \sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}\neq 0}^{n+2} \gamma_i \delta \overline{I}_{i,j} Q_{\overline{W}_j} \bmod p$$

Note that

$$\begin{aligned} E_i &= \hat{I}_i \cdot \hat{Q}_{\overline{W}} \\ &= \sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}\neq 0}^{n+2} (\gamma_i \delta \overline{I}_{i,j} Q_{\overline{W}_j} + \gamma_i t_j \overline{I}_{i,j} + m_j \delta Q_{\overline{W}_j} + m_j t_j) + \\ &\quad \sum_{j=1, I_{i,j}=0, Q_{\overline{W}_j}\neq 0}^{n+2} (m_j \delta Q_{\overline{W}_j} + m_j t_j) + \sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}=0}^{n+2} (\gamma_i t_j \overline{I}_{i,j} + m_j t_j) + \\ &\quad \sum_{j=1, I_{i,j}=0, Q_{\overline{W}_j}=0}^{n+2} m_j t_j \\ &= E_i' + \rho \mod p \end{aligned}$$

If $E_i < p$ and $\rho < \gamma_i \delta$ hold, then we have

$$\begin{aligned} Score(\overline{I}_i, Q_{\overline{W}}) &= \frac{E_i - (E_i \bmod \delta \cdot \gamma_i)}{\delta \cdot \gamma_i} \\ &= \frac{E_i - \rho}{\delta \cdot \gamma_i} \\ &= \frac{\sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}\neq 0}^{n+2} (\gamma_i \delta \overline{I}_{i,j} Q_{\overline{W}_j})}{\delta \cdot \gamma_i} \\ &= \sum_{j=1, I_{i,j}\neq 0, Q_{\overline{W}_j}\neq 0}^{n+2} (\overline{I}_{i,j} Q_{\overline{W}_j}) \\ &= \overline{I}_i \cdot Q_{\overline{W}} \mod p \end{aligned}$$

Thus, the proposed scheme is correct.

4 Analysis and Comparison

4.1 Security Analysis

The proposed multi-keyword search scheme can guarantee the privacy of outsourced data in the known ciphertext attack model and the known background attack model.

Resistance of the Known Ciphertext Attacks. In the dada processing phase, the cloud server can get the encrypted documents and ciphertext indexes, while in the search phase, it is given the search trapdoors. These information are submitted to the cloud server in ciphertext format, where one-time parameters are used for processing each document, index and search query. Thus, the cloud server cannot deduce any sensitive information of outsourced documents and the private key of data owner even though it holds a lots of outsourced materials.

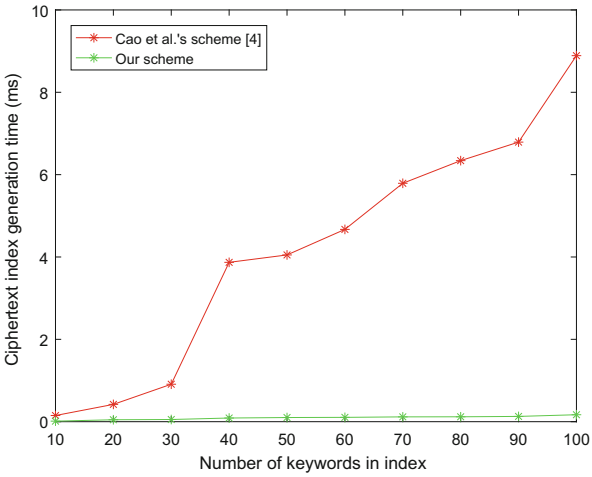

Fig. 2. Time cost on ciphertext index generation.

Resistance of the Known Background Attacks. Under this type of attacks, the cloud server can also get some background information of outsourced documents, for example, keyword frequency. In our scheme, for generating a search trapdoor, the one-time elements t_j and δ are randomly picked, which means that the same search query vector will be mapped to different search trapdoors in different round of searching requests. Moreover, the cloud server is unable to infer the real search query vector from these search trapdoors.

4.2 Performance Evaluation

We conduct experimental evaluation of our scheme and compare with Cao et al.'s scheme [4]. The experiments are implemented using Matlab on a Windows 10 operation system with Intel(R) Core(TM) i5-6500 Processor 3.20 GHz and 8 GB memory. In experiments, we compare the performance of each procedures, that is, ciphertext index construction, search trapdoor generation and cloud search. In experiments, the parameters satisfy $n \leq 2^{32}$, $|\gamma_i| = |\delta| = \lambda_1 = 200$, $|p| = \lambda_2 = 512$, $|m_j| = \lambda_3 = 128$, and $|t_j| = \lambda_4 = 128$.

As shown in Fig. 2, we set the size of the vector from 10 to 100 to evaluate the performance of generating ciphertext index. It can be seen that the time costs of our scheme are less than 1ms for all cases, while the costs of Cao et al.'s scheme [4] rapidly increase as the number of keywords increases. As shown in Fig. 3, the time costs of both schemes are linear with the number of keywords in the query. Note that Cao et al.'s scheme [4] needs to perform matrix multiplications in generating search trapdoor. Thus, our scheme is more efficient than their scheme in all cases. For the search by the cloud server, our scheme does not involve complicated computation operations. Thus, as shown in Fig. 4, the performance of keyword search of our scheme keeps roughly the same for all cases. Whereas

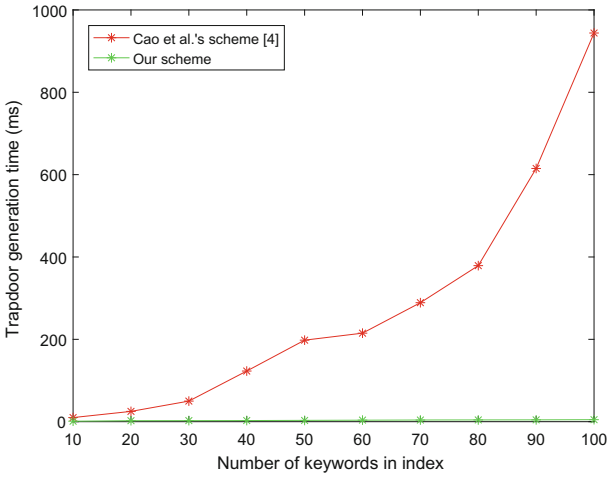

Fig. 3. Time cost on search trapdoor generation.

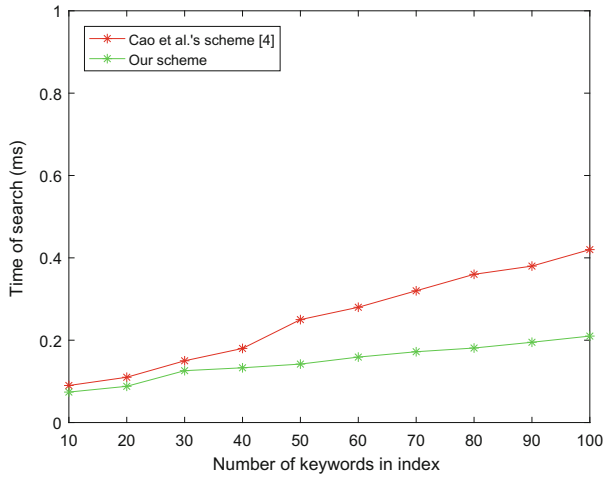

Fig. 4. Time cost on search process.

for Cao et al.'s scheme [4], the performance decreases greatly as the number of keywords in the query vector increasing. Thus, our scheme is more efficient than their scheme.

5 Conclusion

Existing privacy-preserving multi-keyword search schemes cannot be deployed on resource-constrained devices due to the complicated computation operations at

the device side. To address this issue, this paper presented a light-weight multi-keyword search scheme to allow weak device to process data and generate search trapdoors of outsourced documents. Our proposal can protect the privacy of outsourced documents, index and search trapdoor against the known ciphertext attacks and known background attacks. Performance analysis demonstrated that our scheme is more efficient than existing proposals and can be deployed on resource-constrained devices.

Acknowledgements. This article is supported in part by the National Natural Science Foundation of China under projects 61862012, 61772150, 61862011, 61962012 and 61902123, the Guangxi Key R&D Program under project AB17195025, the Guangxi Natural Science Foundation under grants 2018GXNSFDA281054, 2018GXNSFAA281232, 2019GXNSFFA245015, 2019GXNSFGA245004 and AD19245048, the Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, the China Postdoctoral Science Foundation under Project 2019M662769, and the Natural Science Foundation of Hunan Province under Project 2020JJ5085.

References

1. Anand, V., Satapathy, S.C.: Homomorphic encryption for secure information retrieval from the cloud. In: 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), pp. 1–5, February 2016. https://doi.org/10.1109/ICETETS.2016.7602988
2. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_30
3. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30
4. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**(1), 222–233 (2014). https://doi.org/10.1109/TPDS.2013.45
5. Cash, D., Jarecki, S., Jutla, C., Krawczyk, H., Roşu, M.-C., Steiner, M.: Highly-scalable searchable symmetric encryption with support for boolean queries. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 353–373. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_20
6. Chang, Y.-C., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_30
7. Ding, S., Li, Y., Zhang, J., Chen, L., Wang, Z., Xu, Q.: An efficient and privacy-preserving ranked fuzzy keywords search over encrypted cloud data. In: 2016 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC), pp. 1–6, November 2016. https://doi.org/10.1109/BESC.2016.7804500
8. Ding, X., Liu, P., Jin, H.: Privacy-preserving multi-keyword top-k k similarity search over encrypted data. IEEE Trans. Dependable Secure Comput. **16**(2), 344–357 (2019). https://doi.org/10.1109/TDSC.2017.2693969

9. Fu, Z., Wu, X., Guan, C., Sun, X., Ren, K.: Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans. Inf. Forensics Secur. **11**(12), 2706–2716 (2016). https://doi.org/10.1109/TIFS.2016.2596138
10. Fu, Z., Xia, L., Sun, X., Liu, A.X., Xie, G.: Semantic-aware searching over encrypted data for cloud computing. IEEE Trans. Inf. Forensics Secur. **13**(9), 2359–2371 (2018). https://doi.org/10.1109/TIFS.2018.2819121
11. Golle, P., Staddon, J., Waters, B.: Secure conjunctive keyword search over encrypted data. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 31–45. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24852-1_3
12. He, T., Ma, W.: An effective fuzzy keyword search scheme in cloud computing. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 786–789, September 2013. https://doi.org/10.1109/INCoS.2013.150
13. Hu, C., Liu, P., Yang, R., Xu, Y.: Public-key encryption with keyword search via obfuscation. IEEE Access **7**, 37394–37405 (2019). https://doi.org/10.1109/ACCESS.2019.2905250
14. Hwang, Y.H., Lee, P.J.: Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: Takagi, T., Okamoto, E., Okamoto, T., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 2–22. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73489-5_2
15. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Fuzzy keyword search over encrypted data in cloud computing. In: 2010 Proceedings IEEE INFOCOM, pp. 1–5, March 2010. https://doi.org/10.1109/INFCOM.2010.5462196
16. Liu, J., Han, J., Wang, Z.: Searchable encryption scheme on the cloud via fully homomorphic encryption. In: 2016 Sixth International Conference on Instrumentation Measurement, Computer, Communication and Control (IMCCC), pp. 108–111, July 2016. https://doi.org/10.1109/IMCCC.2016.201
17. Liu, Q., Nie, X., Liu, X., Peng, T., Wu, J.: Verifiable ranked search over dynamic encrypted data in cloud computing. In: 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), pp. 1–6, June 2017. https://doi.org/10.1109/IWQoS.2017.7969156
18. Lu, R., Zhu, H., Liu, X., Liu, J.K., Shao, J.: Toward efficient and privacy-preserving computing in big data era. IEEE Network **28**(4), 46–50 (2014). https://doi.org/10.1109/MNET.2014.6863131
19. Raghavendra, S., et al.: IGSK: index generation on split keyword for search over cloud data. In: 2015 International Conference on Computing and Network Communications (CoCoNet), pp. 374–380, December 2015. https://doi.org/10.1109/CoCoNet.2015.7411213
20. Ren, Y., Chen, Y., Yang, J., Xie, B.: Privacy-preserving ranked multi-keyword search leveraging polynomial function in cloud computing. In: 2014 IEEE Global Communications Conference, pp. 594–600, December 2014. https://doi.org/10.1109/GLOCOM.2014.7036872
21. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy, SP 2000, pp. 44–55, May 2000. https://doi.org/10.1109/SECPRI.2000.848445
22. Wang, B., Hou, Y., Li, M., Wang, H., Li, H.: Maple: scalable multi-dimensional range search over encrypted cloud data with tree-based index. In: Proceedings of ACM ASIACCS (2014). https://doi.org/10.1145/2590296.2590305

23. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data. In: 2010 IEEE 30th International Conference on Distributed Computing Systems, pp. 253–262, June 2010. https://doi.org/10.1109/ICDCS.2010.34
24. Xu, P., Tang, S., Xu, P., Wu, Q., Hu, H., Susilo, W.: Practical multi-keyword and boolean search over encrypted e-mail in cloud server. IEEE Trans. Serv. Comput. (2019). https://doi.org/10.1109/TSC.2019.2903502
25. Xu, P., Wu, Q., Wang, W., Susilo, W., Domingo-Ferrer, J., Jin, H.: Generating searchable public-key ciphertexts with hidden structures for fast keyword search. IEEE Trans. Inf. Forensics Secur. **10**(9), 1993–2006 (2015). https://doi.org/10.1109/TIFS.2015.2442220
26. Zhang, L., Zhang, Y., Ma, H.: Privacy-preserving and dynamic multi-attribute conjunctive keyword search over encrypted cloud data. IEEE Access **6**, 34214–34225 (2018). https://doi.org/10.1109/ACCESS.2018.2823718
27. Zhang, Q., Fu, S., Jia, N., Tang, J., Xu, M.: Secure multi-keyword fuzzy search supporting logic query over encrypted cloud data. In: Li, J., Liu, Z., Peng, H. (eds.) SPNCE 2019. LNICST, vol. 284, pp. 210–225. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21373-2_17