SCIENCE CHINA
Information Sciences

• LETTER •

May 2022, Vol. 65 159102:1–159102:3
https://doi.org/10.1007/s11432-019-2668-x

Forward and backward secure searchable encryption with multi-keyword search and result verification

Jie CUI[1,2,3], Yue SUN[1,2], Yan XU[1,2,3], Miaomiao TIAN[1,2,3] & Hong ZHONG[1,2,3*]

[1]*School of Computer Science and Technology, Anhui University, Hefei* 230039*, China;*
[2]*Institute of Physical Science and Information Technology, Anhui University, Hefei* 230039*, China;*
[3]*Anhui Engineering Laboratory of LoT Security Technologies, Anhui University, Hefei* 230039*, China*

Received 2 July 2019/Accepted 24 September 2019/Published online 26 May 2021

Citation Cui J, Sun Y, Xu Y, et al. Forward and backward secure searchable encryption with multi-keyword search and result verification. Sci China Inf Sci, 2022, 65(5): 159102, https://doi.org/10.1007/s11432-019-2668-x

Dear editor,

Searchable encryption [1] is a new search method that not only protects personal privacy, but also searches over encrypted data. The multi-keyword scheme can search documents with more keywords so that the returned results are accurate. Wang et al. [2] presented a multi-keyword search scheme based on the inverted index to improve search efficiency. To obtain complete files from the cloud, Yin et al. [3] proposed the search scheme with result verification. In this scheme, the server can verify the correctness of each file in the results.

In the recent studies, forward and backward securities have received widespread attention. The forward security is that the server cannot know whether the newly added file contains keywords searched before. In the delete process, the backward security cannot leak the information about the keyword deleted to the server. Bost et al. [4] proposed the concept of backward security and constructed the backward security scheme using the range constrained and puncturable pseudorandom functions (PRF). If the data owner wants to delete keywords in a file, the secret key about the keywords should be updated by the PRF algorithm. Because of the PRF algorithm based on the public-key algorithm, the search efficiency is low. Sun et al. [5] improved search efficiency based on Bost et al.'s scheme with the symmetric puncturable encryption (SPE). Because the symmetric encryption algorithm is used replacing PRF in this scheme, it improves search efficiency.

The existing schemes of backward security only support single-keyword search. To solve the problem, we propose a novel searchable encryption scheme that supports multi-keyword search and result verification while satisfies both forward and backward security requirements. Our scheme not only enhances search functioning but also achieves a higher level of security.

Model and algorithm. The system mainly has four entities: cloud server (CS), third-party audit (TPA), data owner (DO), and data user (DU). Firstly, the data owner builds the index according to the keywords $W = \{w_1, w_2, \ldots, w_m\}$ extracted from the file set $D = \{f_1, f_2, \ldots, f_n\}$. Then, the searchable encryption algorithm is used to generate the index I and the symmetric encryption algorithm AES is used to encrypt the file set D to C. Besides, it computes the signature sig for each file. Finally, the encrypted index $I = \{I_1, I_2, I_3\}$ and ciphertext $C = \{C_1, C_2, \ldots, C_n\}$ are uploaded to the CS, and the signature $\mathrm{Sig} = \{\mathrm{sig}_1, \mathrm{sig}_2, \ldots, \mathrm{sig}_n\}$ is sent to the TPA.

(1) Setup. In this algorithm, we choose a security parameter λ. And we also select one-way hash function, H_1, H_2, H_3, H_4, H_5: $\{0,1\}^* \to Z_p^*$. Besides, we also select a bilinear group G of prime order p and the generator is g. Select random number $r \in Z_p^*$ and compute public key g^r. Then, calculate the master secret key MSK for each keyword in dictionary and initialize the puncturable secret key PSK to empty. The public/secret key is (pk,sk), where $\mathrm{pk} = g^r$, $\mathrm{sk} = r$. The k_1, k_2: $\{0,1\} \to \{0,1\}^\lambda$ are the secret key to encrypt keyword and file, respectively.

(2) Build index. Index Generation$(W, \mathrm{SK}) \to I$: When DO uploads the data to cloud, the index should be generated for all files. The keyword set $W = \{w_1, w_2, \ldots, w_m\}$ is extracted from the file set $D = \{f_1, f_2, \ldots, f_n\}$, where m and n indicate the number of all keywords and the all files, respectively. For each keyword w_i $(1 \leqslant i \leqslant m)$, DO should obtain the file set $F = \{f_1, f_2, \ldots, f_t\}$, which each file contains the keyword w_i, where t indicates the number of file. Then, DO generates the index below. Firstly, the algorithm calculates the equation, $H_1(\mathrm{Enc}_{k_1}(w_i||j))$ and $H_2(\mathrm{Enc}_{k_1}(w_i||j)) \oplus \mathrm{Enc}_{k_1}(w_i||j-1)$, where $1 \leqslant j \leqslant t$. In the process of iteration, it stores $I_1[H_1(\mathrm{Enc}_{k_1}(w_i||j))] = H_2(\mathrm{Enc}_{k_1}(w_i||j)) \oplus (\mathrm{Enc}_{k_1}(w_i||j-1))$ in the index I_1 and $I_2[H_3(\mathrm{Enc}_{k_1}(w_i||j))] = (\mathrm{Enc}_{k_t}(\mathrm{ind}_j), t)$ in the index I_2, where ind_j is the identifier of file f_j, $k_t = \oplus_{i=1}^{d} F(\mathrm{sk}_i, t)$, $t = F(w_i, \mathrm{ind}_j)$. In addition, it generates the new data structure Boolean Data to judge whether the keyword belongs to a file. It calculates tag $t = F(\mathrm{Enc}_{k_1}(w_i), \mathrm{ind}_j)$ and secret key $\mathrm{sk}_h = H(\mathrm{sk}_{h-1}, h)$ $(0 \leqslant h \leqslant d)$, where the sk_0 and d from the master secret key $\mathrm{msk}_{w_i} = (\mathrm{sk}_0, d)$. And

* Corresponding author (email: zhongh@ahu.edu.cn)

(a)

Scheme	Multi-keyword	Forward security	Backward security	Search efficiency
Wang et al. [2]	√	×	×	$O(n)$
Wu et al. [6]	√	√	×	$O(\log n)$
Our	√	√	√	$O(r)$

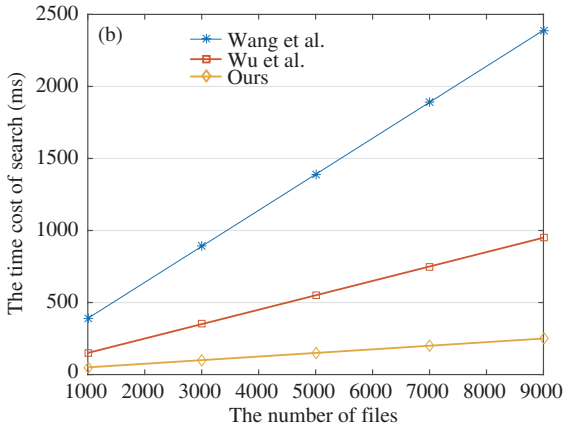

Figure 1 (Color online) (a) Comparison of related schemes; (b) search time VS the number of documents.

with t and sk_h, it can compute v, $v = \oplus_{h=0}^{d} F(\mathrm{sk}_h, t)$. It stores $I_3[H_4(v)] = \text{true} \oplus v$ in the index I_3.

(3) Generate query trapdoor. When the data user (DU) wants to search for file in the cloud with keywords $\Delta = \{w_1, w_2, \ldots, w_t\}$, the trapdoor should be computed. First, DU obtains the access permission from DO. Second, DU chooses a center keyword w from the Δ, and obtains the count i of w from DO, in which i is the number of file that contains the keyword w. DU calculates $e' = \mathrm{Enc}_{k_1}(w_1||i)$, where assume w_1 is the center keyword. Third, DU uses the symmetric key k_1 to encrypt each keyword in Δ, $e_j = \mathrm{Enc}_{k_1}(w_j)$ $(2 \leqslant j \leqslant t)$, and gets the master secret key msk for each search keyword from DO. Next, the algorithm outputs the trapdoor. Finally, DU sends the T_Δ to CS.

(4) Search process. When CS receives the trapdoor T_Δ from DU, CS will perform the search algorithm. First, CS uses the e' to find all address of ciphertext in index I_2, and computes the equation as $\mathrm{Enc}_{k_1}(\mathrm{w}_1||i-1) = I_1[H_1(\mathrm{Enc}_{k_1}(w_1||i))] \oplus H_2(\mathrm{Enc}_{k_1}(w_1||i))$, where $e' = \mathrm{Enc}_{k_1}(w_1||i)$. In the process, CS can get the all address about keyword w_1 in I_2, and utilize the address to get all ciphertext $(e,t) = I_2[H_3(\mathrm{Enc}_{k_1}(w_1||j))]$ $(1 \leqslant j \leqslant i)$, where the i is the number of file that contains the keyword w_1. Then, CS computes the k with t, $k = \oplus F_{i=1}^{l}.\mathrm{Eval}(\mathrm{psk}_i, t) \oplus \oplus_{j=l}^{d} F(\mathrm{sk}_j, t)$. And CS uses the secret key k to decrypt e with symmetric encrypt algorithm, $\mathrm{ind} = \mathrm{Dec}_k(e)$, and saves identifier ind in Res in which the file contains the keyword w_1. Finally, all files are contained in Res, which includes the search keyword w_1. For each $\mathrm{ind}_i (1 \leqslant i \leqslant \|\mathrm{Res}\|)$ in Res, CS calculates the tag t and hash value v, $t = F(e_j, \mathrm{ind}_i)$ and $v = \oplus_{s=1}^{l} F.\mathrm{Eval}(\mathrm{psk}_{e_{j_s}}, t) \oplus \oplus_{m=l}^{d} F(\mathrm{sk}_{e_{jm}}, t)$, where e is from the trapdoor T_Δ, the puncturable secret key psk is from the PSK in cloud, and the master secret key msk is from the trapdoor, $2 \leqslant j \leqslant t$. In the process, the each pair ind and e can get t to calculate the value v. Then, CS gets a vable b with v, $b = I_3[H_4(v)] \oplus v$. If b is true, it indicates the file f_i containing the keyword w_j. Otherwise, or not. In the process, when a file contains all search keywords in T_Δ, it will be retained in Res. Finally, CS finds the ciphertext over the database by the identifier in Res. In the end, CS sends CT to TPA.

(5) Verification. The result CT will be checked by the TPA before it is returned to DU. When TPA receives the CT from CS, it will check the integrality of files in CT. For each file C_i in CT, CS calculates the equation:

$$e(\mathrm{sig}_i, g) = e(H_5(\mathrm{ind}_i) \cdot g^{H_5(C_i)}, \mathrm{pk}),$$

where g is the generator, pk is the public key. If the equation holds, the file C_i is complete and saved in CT. Otherwise, TPA will discard the file C_i from CT. In the end, TPA sends the CT to the DU.

(6) Update process. Add Data: DO wants to add the data into the cloud. First, DO will generate the sub-index $I' = \{I'_1, I'_2, I'_3\}$ with the algorithm of index generation. Then, DO uses AES and signature algorithm to compute C and sig, respectively. Finally, DO sends the index I', ciphertext C to CS and signature sig to TPA.

Delete Data: DO wants to delete the data that is (w, ind_i) from the file f_i, where file f_i contains the keyword w. DO uses the symmetric puncturable encryption (SPE) algorithm to puncture the secret key of keyword w. First, DO calculates the tag $t = F(\mathrm{Enc}_{k_1}(w), \mathrm{ind}_i)$ for the deleted keyword. Then, DO uses the tag t to calculate the punctured key $\mathrm{psk} = F.\mathrm{Punc}(\mathrm{sk}, t)$, where the $F.\mathrm{Punc}$ is the puncture algorithm and the sk is the secret key of the keyword w from MSK[w]. Finally, DO sends the psk to CS.

We choose the request for comments as the file set. In our experiment, we select 9000 files from the file set RFC with 5000 keywords. Each file extracts the keyword from 90 to 120. The programming language is Java 7 and the machine is Windows 7 system 64 with an Intel Core i5-4440 CPU (3.10 GHz, 8 GB RAM). In the experiment, we use the AES algorithm as the CPA secure symmetric encryption algorithm and the SHA-256 as the hash function. The search time is shown in Figure 1(b). Our scheme has the fewest time than that of the related scheme. Because our scheme uses the two-stage index to search documents in the search process, it reduces the search on documents that cannot contain the search keywords to achieve optimal search efficiency. The search complexity is $O(r)$, which r is the number of files containing the search keywords. In Wang et al.'s scheme, the server searches files on the entire file set. The search complexity is $O(n)$. In Wu et al.'s scheme [6], the search based on the tree index improves search efficiency. The search efficiency is $O(\log n)$.

Conclusion. In this study, we proposed a novel forward and backward secure symmetric search encryption (SSE) scheme that supports multi-keyword searches and result verification. Compared to existing schemes, the proposed scheme is more practical in terms of its application. In addition, the scheme supports a result verification process, which can discard useless results and increase the utilization rate of the network. During the search process, the data user receives files that are complete and precise, which

enhances the practicability of the scheme. Therefore, the scheme not only has a powerful search capability, but also a higher security level. In addition, in the further research, we will consider blockchain technology and artificial intelligence technology to support the SSE scheme, which can achieve optimal search efficiency.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61572001, 61872001) and Special Fund for Key Program of Science and Technology of Anhui Province (Grant No. 18030901027).

References

1 Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In: Proceedings of Security and Privacy, 2000. 44–55

2 Wang B, Song W, Lou W, et al. Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee. In: Proceedings of IEEE Conference on Computer Communications, 2015. 2092–2100

3 Yin H, Qin Z, Zhang J X, et al. Achieving secure, universal, and fine-grained query results verification for secure search scheme over encrypted cloud data. IEEE Trans Cloud Comput, 2017. doi: 10.1109/TCC.2017.2709318

4 Bost R, Minaud B, Ohrimenko O. Forwardand backward private searchable encryption from constrained cryptographic primitives. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2017. 1465–1482

5 Sun S F, Yuan X, Liu J K, et al. Practical backward-secure searchable encryption from symmetric puncturable encryption. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2018. 763–780

6 Wu Z Q, Li K L. VBTree: forward secure conjunctive queries over encrypted data for cloud computing. Int J Very Large Data Bases, 2019, 28: 25–46