A Data Sorting and Searching Scheme Based on Distributed Asymmetric Searchable Encryption

Lina Zou, Xueying Wang and Shoulin Yin
(Corresponding author: Shoulin Yin)

Department of Computer and Mathematics, Shenyang Normal University
253 Huanghe N. St, Huanggu Qu, Shenyang 110034, China
(Email: zln0781@sina.com)
(Received Dec. 25, 2016; revised and accepted Mar. 12 & May 2, 2017)

Abstract

Searchable encryption algorithm is a hot issue nowadays. It can sort the results of searching and return the optimal matching files. The essence of Asymmetric searchable encryption is that users exchange the data of encryption, one party sends a ciphertext with key encryption, the other party with another key receives the ciphertext. Encryption key is not the same as the decryption key, and cannot deduce another key from any one of the key, thus it greatly enhances the information protection, and can prevent leakage the user's search pattern. In order to get higher efficiency and security in information retrieval, in this paper we introduce the concept of distributed Searchable asymmetric encryption, which is useful for security and can enable search operations on encrypted data. Moreover, we give the proof of security. Finally, experiments results show that our method has better retrieval efficiency.

Keywords: Asymmetric Searchable Encryption; Data Sorting; Distributed; Searchable Encryption

1 Introduction

With the rapid development of communication technology, cloud service has entered the large number of people's live and work. Exposing the user data security of the third party service providers leads to security issues [12, 28]. To protect user's data privacy has become more and more important and urgent, which requires encryption. However, the cloud service that its characteristics of convenient and flexible way to charge, more and more users choose the local data migration to the cloud server. Many netters delegate to a third party provider or service provider right to search for their data. There are many scholars having done some research about the Public Key Encryption with Keyword Search. For example, sensitive data (i.e. private data of user or commercial data) is put into cloud, cloud service providers can directly read and use these data, which may result in bad effect of violating the privacy of users and damaging the security of data [7, 19, 20, 22, 23]. Therefore, the cloud service is not absolutely reliable. We need to use new technologies to protect privacy and data. And on the premise of guaranteeing safety, it needs to safeguard the normal operation of user as soon as possible. So protection of user privacy and safety of user data becomes a hot issue.

To solve this problem, Searchable Encryption is introduced [2, 8, 9]. Using SE mechanism encrypts data, and the ciphertext is stored in the cloud server. When users need to search some keywords, they can use the keyword to search documents sent to the cloud server [13]. The cloud will receive the search proof test matching for each file, if the match is successful, it implies that the file contains the keyword. Finally, the cloud will return all files matching success back to the user. After receiving the search results, users only need to return to the encrypted files. The majority of the schemes study single keyword, conjunctive keywords and complex search query of public key cryptography based SE schemes [4, 5, 16].

Here is a motivating example for PEKS. This example is according to the reference [1, 15, 24]. Suppose user Alice wants to read her emails from her laptop or smart phone or PAD after she stores her emails in the servers of some email service provider. Because of previous cloud accidents, Alice does not believe the third-party service provider or fears that powerful agencies may require the service provider to surrender all her data. Any user with Alice's public key can send her encrypted emails from the many transmission mediums that only she can decrypt based on standard public key encryption. PEKS scheme produces some email searchable ciphertexts, Alice prepare to find a unique email then, the sender could also attach to the searchable ciphertexts. Alice could make use of keywords to search for this email. Once delegated, the ciphertexts can be searched. Across Alice's email the service provider searches those search ciphertexts contained that match the issued trapdoor, and returns to her a positive match.

There is a standard application in searchable encryption that it supports the order with keyword matching degree in returned document. Hwang [10] proposed an efficient secure channel free public key encryption with conjunctive field keyword search scheme that could stand against the off-line keyword-guessing attacks, which was more suitable for the weak devices used by users. Tsai [25] proposed a publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms. He pointed out that even if either factoring or discrete logarithms was broken, this scheme still could keep the authentication, integration, and confidentiality of the message. Ling [14] proposed an efficient and secure onetime password authentication scheme for wireless sensor networks according to the Lamport's concept to consider the limitations of computation and lower power in a wireless sensor networks. Cao [3] defined and solved the challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE). A set of strict privacy requirements are established for such a secure cloud data utilization system. Among various multikeyword semantics, it chose the efficient similarity measure of "coordinate matching" to capture the relevance of data documents to the search query. They further used "inner product similarity" to quantitatively evaluate such similarity measure and proposed a basic idea for the MRSE based on secure inner product computation, and then gave two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. Wang [26] used order preserving encryption to encrypt relevance, which could get accuracy results. Wang [27] presented that ranked search greatly enhanced system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensured the file retrieval accuracy. Specifically, he explored the statistical measure approach, i.e. relevance score, from information retrieval to build a secure searchable index, and developed a one-to-many order-preserving mapping technique to properly protect those sensitive score information. Jiang [11] proposed a novel privacy preserving keyword search scheme over encrypted cloud data to address this problem. To enable users to search over encrypted data, he firstly adopted a structure named as Inverted Matrix (IM) to build search index. The IM was consisted of a number of index vectors, each of which was associated with a keyword. Then he mapped a keyword to a value as an address used to locate the corresponding index vector. Finally, he masked index vectors with pseudo-random bits to obtain an Encrypted Enlarged Inverted Matrix (EEIM) to preserve the privacy of users. However, the above methods are used for symmetric searchable encryption or some asymmetric searchable encryption methods have low efficiency.

Therefore, we propose a data sorting and searching based on distributed asymmetric searchable encryption in cloud environment. And we combine order-preserving encryption algorithm based on symmetric encryption to realize sorting and searching in asymmetric searchable encryption algorithm. The following is the structure of this paper. In Section 2, we construct the new scheme. Section 3 and Section 4 give the security proof and performance analysis respectively. There is a conclusion in Section 5.

2 Structure of Distributed Asymmetric Encryption Algorithm

Firstly, we introduce distributed asymmetric encryption algorithm (DAEA). A DAEA system includes four probabilistic polynomial time algorithms as follows:

- $KeyGen$: $(K_C, K_{SP_1}, K_{SP_2}, K_{QP_1}, K_{QP_2})$. This algorithm is executed by the client C, inputs a security parameter λ and outputs a secret key K_C to the client C, public keys K_{SP_1} and K_{QP_1} to SP and QP ($K_{SP_1} = K_{QP_1}$) and private keys K_{SP_2} and K_{QP_2} to SP and QP, respectively.

- $Encrypt$: $(I_1, I_2) = Encrypt(K_C, D)$. This algorithm is executed by the client C, inputs a key KC and a set of documents D, outputs an encrypted index I_1 to SP and I_2 to QP.

- $Trapdoor$: (T_1^s, T_2^s). This algorithm is executed by the client C, inputs the private key K_{SP_2}, K_{QP_2} and a query keyword $s \in W$, and outputs a trapdoor T_1^s to SP and trapdoor T_2^s to QP.

- $Test$: $(a = Test(K_{SP_1}, K_{QP_1}, I_1, I_2, T_1^s, T_2^s))$. SP provides K_{SP_1}, I_1, T_1^s and QP provides K_{QP_1}, I_2, T_2^s as input. According to the matching results of W and W' outputs judgment value a, $a \in 0, 1$.

Given the above definition, the public-key encryption scheme with keyword search does the following processes. Firstly, the receiver uses the Setup algorithm to produce his/her public or private key pair. Then, he/she runs the Trapdoor algorithm to create trapdoors T_W (the third service providers can search) for any keyword W. The given trapdoors are as input for the Test algorithm. The third service provider determines whether one sender gives message encrypted by ksEnc containing one of the keywords W specified by the receiver.

Sorting and searching function indicates that all the matched documents will be ordered by a standard. Finally, it returns the most relative k documents to user. Its SQL form is "**ORDERED BY keyword**". We use the order-preserving encryption algorithm to compute correlation score. Sorting function can record encrypted correlation score and construct an index $< keyword, score >$ Key-value pair. Therefore, sorting function can acquire score and order with computing time complexity $O(1)$.

To store index record, this paper utilizes indirect addressing scheme [18, 21] based on sparse matrix to construct a 2-dimension index table A and record $< keyword, correlationscore >$. All the data are encrypted.

When it executes query, server searches all the correlation score of matched documents and selects the optimal k documents. Before encrypting data, it needs a preprocess to safety use order-preserving encryption algorithm. So we construct a order–preserving encryption table to preprocess all the plaintexts. The followings are the steps to store encrypted correlation score.

1) Given a document set $D = (d_1, d_2, \cdots, d_n)$. It scans o^k keywords W for each document $d_k (1 \le k \le n)$ in document set. In d_k, each keyword $w_i^k \in W$. According to the appear frequency of keyword, it calculates correlation score $s_i^k (1 \le i \le o^k)$ and records a $o^k \times 3$ matrix corresponding to d_k. In this matrix, the record of $i-th$ row is $R_i^k = (w_i^k, s_i^k, p_i^k)$.

2) For all documents, data quantity of order-preserving encryption is $N = o^1 + o^2 + \cdots + o^n$ and numbered $s_1, s_2, \cdots, s_N$. Order-preserving encryption result of each data $s_j (1 \le j \le N)$ is e_j. The previous matrix is transformed into a order-preserving encryption table, $i-th$ row records $R_i^k = (w_i^k, s_i^k, p_i^k)$, where e_i^k is the order-preserving encryption result of s_i^k.

3) For a document, it contains $\frac{|d|}{2} + 1$ keywords at most (there is a separator behind each keyword). Therefore, the index table will be filled with $\frac{|d|}{2} + 1$ data items at last, which guarantees that document content has nothing to do with data item number.

The definitions in asymmetric searchable encryption algorithm will be used in this paper as follows.

Definition 1. $s \leftarrow PEKS(K_{pub}, w)$. *Construction subalgorithm of searchable structure. Input public key K_{pub} and a keyword w. Output searchable structure s. This algorithm is executed by user. s and encryption information of document will be submitted to server.*

Definition 2. $c \leftarrow Enc(K_{pub}, d)$. *Document encryption subalgorithm of asymmetric searchable. Input public key K_{pub} and a message d. Output ciphertext c. This algorithm is executed by data sender. c and searchable encryption structures will be submitted to server.*

Definition 3. *Hash function $f_h : 0,1^* \to 0,1^l$. Where l is mapping length. For example, $f_h = MD5$, then $l = 128bits$.*

Our new data sorting and searching scheme based on distributed asymmetric searchable encryption includes two parts: Build algorithm and Filter algorithm. First, it inputs one document d and encryption key in $Build(d, K_{pub})$ algorithm. Second, it scans d and constructs words list. Third, keywords are compared. Fourth, it outputs functional structure and word mapping. The detailed processes of two algorithms are used for encryption function construction and file query respectively as follows (**Algorithm 1** and **Algorithm 2**).

The main idea of Build algorithm is that it extracts keywords from document to construct index table combining data structure of asymmetric searchable encryption

Algorithm 1 Build algorithm $Build(d, K_{pub})$

Input: Document d, encryption public key $K_e = K_{pub}$.
Output: Function structure $L_d = A$ and word mapping $V_d = (v_1, v_2, \cdots, v_r)$.

1) Compute $c \leftarrow ASE.Enc(K_{pub}, d)$.
2) Scan d and get r words. Construct a word table $W = (w_1, w_2, \cdots, w_r)$.
3) Select a different keywords $W' = (w_1, w_2, \cdots, w_a)$ from W.
4) for each word $w_x (1 \le x \le a)$ in W' do
5) Compute $s_x \leftarrow ASE.PEKS(K_{pub}, w_x)$
6) Compute $h_x \leftarrow f_h(s_x)$
7) end for
8) Let $G = (s_1, s_2, \cdots, s_a)$ map to $H = (h_1, h_2, \cdots, h_a)$ and W'
9) for each word $w_y (1 \le y \le r)$ in W do
10) search $h \in H_i$
11) Let $v_y = h$
12) end for
13) Let $V_d = (v_1, v_2, \cdots, v_r)$
14) Data item of document d in order-preserving encryption table is (w_1, e_1, p_1), (w_2, e_2, p_2), $\cdots$, (w_o, e_o, p_o)
15) for each $i \in [0, 1]$, build an index $A[v_{p_i}] = e_i$
16) the rest $\frac{|d|}{2} + 1$ items are filled with random character string.
17) Output.

Algorithm 2 Filter algorithm $Filter(C, L, V_T)$

1) Input n ciphertexts $C = (c_1, c_2, \cdots, c_n)$, corresponding function structure $L = (L_1, L_2, \cdots, L_n)$, mapping $V_T = (V_1, V_2, \cdots, V_n) = (v_1, v_2, \cdots, v_n)$
2) for n function structures, compute $r_1 = L_1[v_1]$, $r_2 = L_2[v_2]$, $\cdots$, $r_n = L_n[v_n]$.
3) Output order $C' = (c_1, c_2, \cdots, c_k)$.

algorithm based on order-preserving encryption scheme. And Filter algorithm, according to encryption index table of each encryption document, ranks the query result and returns the order result. So it can return the optimal match query document.

3 Security Proof

For SP and QP, we suppose secure channels between the three parties which does not collude. It indicates that admissible $Q-query$ protocol running $\prod_{DASE}^{Q}$ $(Q \in N)$ are executed. Normally, to all participants, the protocol $\prod_{DASE}^{Q}$ has the unique public output access pattern $(id_{w_1}(D), \cdots, id_{w_Q}(D))$. If a DSAE scheme is secure, it leaks no information. The following is that we first define ideal functionality of a DSAE scheme:

Functionality X_{DASE}^{Q}. Consider a DSAE scheme with keyword set W, output $(K_C, K_{SP_1}, K_{SP_2}, K_{QP_1}, K_{QP_2}) = Keygen(\lambda)$, and a document set D. X_{DASE}^{Q} $(Q \in N)$ is the functionality that takes as input:

- K_C and keywords $w_1, \cdots, w_Q$ from client C.
- K_{SP_1}, K_{SP_2} from provider SP.
- K_{QP_1}, K_{QP_2} from query proxy QP.
- $id_{w_1}(D), \cdots, id_{w_Q}(D) \rightarrow id_Q(D)$ to all C, SP, QP.

Then, we consider that a DSAE scheme is secure if all the admissible $Q-query$ run $\prod_{DASE}^{Q}$ to compute functionality X_{DASE}^{Q}.

Safety protection model of sorting query function is that given two same documents set D_1 and D_2. Challenger uses LSE to encrypt data D_b. Adversary can query a keyword and acquire ordered document subset, nevertheless, he dose not know which document subset is chosen by challenger.

According to the proposed non-adaptive indistinguishability chosen keyword attack model [6] and order-persevering concept, we present a non-adaptive indistinguishability chosen order attack model as follows.

Definition 4. ***non-adaptive indistinguishability chosen order attack model****. Let Σ be the order query function component. $k \in N$ is secure parameter. Considering the following simulation experiment. A denotes adversary. S is simulator.*

1) $Adv_{\sum,A}(k)$: challenger executes key generation algorithm $Gen(1^k)$ to generate a key K. Adversary generates a document set $D = (d_1, d_2, \cdots, d_n)$ (size of each document is constant), he receives the encrypted document $C = (c_1, c_2, \cdots, c_n)$ and function structure $L = (L_1, L_2, \cdots, L_n)$. Adversary submits a query w, where $w \in d_1 \cap d_2 \cap \cdots \cap d_n$ and receives mapping v from challenger. At last, A returns $b \in [0, 1]$ as the output of experiment.
2) $Simulate_{\sum,A,S}(k)$. Given documents number n, size of each document $|d|$ and mapping size $|v|$. S generates C^*, L^*, v^* and sends the results to adversary A. A returns $b \in [0, 1]$ as the output of experiment.

Then we call that order query function component is CRKA secure. Only for all polynomial time adversary A, there is a simulator S meeting the following formula:

$$|Pr[Adv_{\sum,A}(k) = 1] - Pr[Simulate_{\sum,A,S}(k) = 1]| \leq negl(k)$$

Its probability depends on key generation algorithm $Gen(1^k)$.

Theorem 1. *If interface of LSE has CPA security, OPE algorithm has the POPF-CCA security, then order query function component has CRKA security.*

Proof. Simulator generates (C^*, L^*, v^{ast}) through the following steps. For C^*, simulator generates n random character string $(c_1^*, c_2^*, \cdots, c_n^*)$. Size of each string is $|d|$. For L^*, let $m = |d|/2 + 1$, simulator generates m random character string $(v_1^*, v_2^*, \cdots, v_m^*)$. Size of each string is $|v|$. Simulator constructs a $m \times n$ matrix $E_{m \times n} = (e_{ij}^*)$. Where e_{ij}^* is random number. For every document, simulator generates index item $A_j^*[v_i^*] = e_{ij}^* (1 \leq i \leq m, 1 \leq j \leq n)$. For v^*, simulator randomly selects $v^* = v_i^* \in V^*$. □

Theorem 2. *There is no polynomial resolving device which can distinguish (C, L, v) and (C^*, L^*, v^*).*

Proof. Key K is encrypted for adversary, hence CPA security of interface directly guarantees the indistinguishability between C and C^*, as well as v and v^*. Meanwhile, when receiving $v = v_i \in V$ or $v^* = v_i^* \in V$, adversary can call the function $Filter(C, L, v)$ or $Filter(C^*, L^*, v^*)$ to acquire $r_1 = L_1[v_i] = (e_1, e_2, \cdots, r_n = L_n[v_i] = e_n)$ or $r_1^* = L_1^*[v_i^*] = (e_1^*, e_2^*, \cdots, r_n^* = L_n^*[v_i^*] = e_n^*)$. It is undistinguish between POPF-CCA security guarantee set $(r_1, r_2, \cdots, r_n)$ and set $(r_1^*, r_2^*, \cdots, r_n^*)$. That is to say, adversary cannot distinguish the encryption result of OPE and output of randomly order-preserving function. Therefore, L and L^* are indistinguishable. □

4 Performance Analysis

This new scheme is coded by MATLAB. Each document is set as 10KB. Its content is the word combination randomly selected from dictionary. Running results of Filter algorithm are as Figure 1.

From Figure 1, we can know that the reason why order asymmetric searchable encryption scheme has high running efficiency is that asymmetric searchable encryption structure is transformed into symmetric searchable encryption structure when encrypting data. They can have the same encryption efficiency. For order query, its main operation is to acquire correlation score from index table. This table is maintained by indirect address of spare

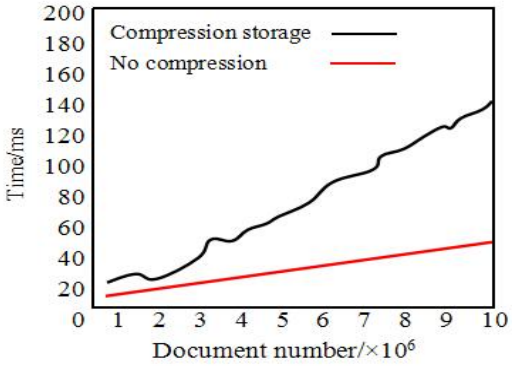

Figure 1: Document retrieval performance

matrix (i.e. compression storage). Let n be number of further required documents. So the total query time complexity is $O(n)$. In addition, we compare the score from the final result and select the optimal k matched documents. Its total computation complexity is $O(n)$ too.

The following is the comparison experiment. The experience data consists of 200 short messages. For the random input keywords set, cloud server will search all the data and find the required keywords set. This experiment is conducted 100 times. And we use three aspects to evaluate the performance of our method including index time cost (Time cost on data owner building retrieval index includes time of extracting and encrypting each keyword in data document.), trapdoor generated time cost (Time cost on data user building trapdoor includes the time of encrypting time and the time of generating trapdoor.) and query time cost (Time cost on cloud server completing a retrieval request includes time of computing document similarity degree and ranking time.).

The document number ranges from 100 to 600, and we select 60 keywords in each document. Table1 is the index time cost with different documents. And a comparison to PRMSN [29] and NMRSS [17] is shown in Table 1. From Table 1, we can know that the time cost of creating index will increase with the adding of documents. In addition, because the time of building sub-index for each document is unchanged, the relation between time cost and document number is quasi-linear.

Table 1: Index time cost with different documents

Document number	The Proposed Method	PRMSN	NMRSS
100	0.25s	0.35s	0.34s
200	0.49s	0.52s	0.51s
300	0.74s	8.79s	0.81s
400	0.93s	1.13s	1.11s
500	1.07s	1.22s	1.22s
600	1.61s	1.71s	1.77s

Then we change the keywords number in each document. The total number of document is 600 under the different keywords. So the index time with different keywords is as shown in Table 2. Table 2 shows the effectiveness of our new method. The more keywords are, the retrieval time is higher. However, our method can take less time.

Table 2: Index time cost with different keywords

Keyword Number	The Proposed Method	PRMSN	NMRSS
10	0.47ms	0.57ms	0.58ms
20	1.12ms	1.33ms	1.30ms
30	1.35ms	1.53ms	1.54ms
40	1.57ms	1.69ms	1.75ms
50	1.74ms	1.92ms	1.89ms
60	1.77ms	1.92ms	1.92ms

Furthermore, we make experience for generating index time with different keyword number as Table 3. In Table 3, when the maximum number of keyword is $d = 30$, the trapdoor generating time is the optimal with our method. From Table 3 we can know that the keyword number cannot affect the time cost and all the time cost is less than 0.002s, which on account of the fact that it adds virtual keywords into the input keyword set. This can ensure that the total number of keyword is d. And Table 4 is the index generating time with different maximum keyword number. Input keyword number is 10 in Table 4 whose time change trend is closely to Table 3. In that keyword number increases, the order of polynomial function increases too, our method costs less time than other two methods.

Table 3: Generating trapdoor time with different keywords number

Keyword Number	The Proposed Method	PRMSN	NMRSS
5	1.14ms	1.40ms	1.42ms
10	1.14ms	1.44ms	1.43ms
15	1.17ms	1.40ms	1.38ms
20	1.17ms	1.40ms	1.40ms
25	1.17ms	1.43ms	1.44ms
30	1.17ms	1.44ms	1.44ms

Table 4: Generating trapdoor time with different maximum keywords number

Maximum Keyword No.	The Proposed Method	PRMSN	NMRSS
10	0.45s	0.79s	0.78s
20	0.94s	1.08s	1.11s
30	1.14s	1.46s	1.47s
40	1.23s	1.57s	1.58s

Finally, we use the inquiry time to demonstrate the

effective of our new method. We first keep that the maximum number of input keyword is 30 and 40 keywords in each document, then we change the document number. Inquiry time with different documents number is as shown in Table 5. Cloud server will cost longer time to search all the data set with the increase of document. Then we set document number as 600 and change the maximum number of input keyword as table6. Data in table6 shows that maximum number of input keyword has none effect on our method. Therefore, our method can execute effectively multi-keyword retrieval in cloud environment.

Table 5: Inquiry time with different documents

Document Number	The Proposed Method	PRMSN	NMRSS
100	0.11s	0.12s	0.13s
200	0.21s	0.21s	0.21s
300	0.28s	0.34s	0.33s
400	0.34s	0.44s	0.43s
500	0.42s	0.54s	0.53s
600	0.62s	0.71s	0.69s

Table 6: Inquiry time with different maximum keywords number

Maximum Keyword No.	The Proposed Method	PRMSN	NMRSS
10	0.57s	0.64s	0.61s
20	0.57s	0.66s	0.62s
30	0.61s	0.68s	0.65s
40	0.78s	0.79s	0.83s

5 Conclusions

In this paper, we propose a data sorting and searching scheme based on asymmetric searchable encryption in cloud environment, which offsets the deficiency of asymmetric searchable encryption. This new scheme combines the advantage of asymmetric searchable encryption and symmetric searchable encryption, it can be extended to other data structure based on symmetric searchable encryption. Therefore, we will study more advanced searchable encryption schemes to improve our method in the future.

6 Acknowledgement

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

References

[1] A. Arriaga, Q. Tang, P. Ryan, "Trapdoor privacy in asymmetric searchable encryption schemes," *International Conference on Cryptology in Africa*, pp. 31-50, 2014.

[2] M. Bellare, A. Boldyreva, A. O'Neill, "Deterministic and efficiently searchable encryption," in *Annual International Cryptology Conference*, pp. 535-552, 2007.

[3] N. Cao, C. Wang, M. Li, K. Ren and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222-233, 2014.

[4] D. Cash, S. Tessaro, "The locality of searchable symmetric encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 351-368, 2014.

[5] Q. Chai, G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *IEEE International Conference on Communications (ICC'12)*, pp. 917-922, 2012.

[6] M. Chase, E. Shen, "Substring-searchable symmetric encryption," *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 263-281, 2015.

[7] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[8] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895-934, 2011.

[9] S. T. Hsu, C. C. Yang, and M. S. Hwang, "A study of public key encryption with keyword search", *International Journal of Network Security*, vol. 15, no. 2, pp. 71-79, Mar. 2013.

[10] M. S. Hwang, S. T. Hsu, C. C. Lee, "A new public key encryption with conjunctive field keyword search scheme," *Information Technology and Control*, vol. 43, no. 3, pp. 277-288, Sep. 2014.

[11] X. Jiang, J. Yu, F. Kong, X. Cheng and R. Hao, "A novel privacy preserving keyword search scheme over encrypted cloud data," in *10th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'15)*, pp. 836-839, 2015.

[12] S. Kamara, K. Lauter, "Cryptographic cloud storage," in *International Conference on Financial Cryptography and Data Security*, pp. 136-149, 2010.

[13] C. C. Lee, S. T. Hsu, M. S. Hwang, "A study of conjunctive keyword searchable schemes," *International Journal of Network Security*, vol. 15, no. 5, pp. 321-330, 2013.

[14] C. H. Ling, C. C. Lee, C. C. Yang and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.

[15] B. Martens, F. Teuteberg, "Risk and compliance management for cloud computing services: Designing a reference model," in *Americas Conference on Information Systems (Amcis'11)*, 2011.
[16] T. Moataz, A. Shikfa, "Boolean symmetric searchable encryption," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 265-276, 2013.
[17] R. R. Netinti, S. Madhri, "A novel multi-keyword ranked search system in encrypted and Synonym queries supported cloud," *International Journal of Science Engineering and Advance Technology*, vol. 3, no. 12, pp. 1370-1373, 2016.
[18] C. Nit, L. M. Itu, C. Suciu, "GPU accelerated blood flow computation using the lattice boltzmann method," in *IEEE High Performance Extreme Computing Conference (HPEC'13)*, pp. 1-6, 2013.
[19] D. Patel, "Accountability in cloud computing by means of chain of trust dipen contractor," *International Journal of Network Security*, vol. 19, no. 2, pp. 251-259, 2017.
[20] S. Subashin, V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1-11, 2011.
[21] C. T. Sungur, P. Spiess, N. Oertel and O. Kopp, "Extending BPMN for wireless sensor networks," in *IEEE 15th Conference on Business Informatics*, pp. 109-116, 2013.
[22] H. Takabi, J. B. D. Joshi, G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24-31, 2010.
[23] H. Takabi, J. B. D. Joshi, G. J. Ahn, "Securecloud: Towards a comprehensive security framework for cloud computing environments," in *IEEE 34th Annual Conference on Computer Software and Applications Conference Workshops (COMPSACW'10)*, pp. 393-398, 2010.
[24] Q. Tang, X. Chen, "Towards asymmetric searchable encryption with message recovery and flexible search authorization," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 253-264, 2013.
[25] C. Y. Tsai, C. Y. Liu, S. C. Tsaur and M. S. Hwang, "A publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms," *International Journal of Network Security*, vol. 19, no. 3, pp. 443-448, May 2017.
[26] C. Wang, N. Cao, J. Li and K. Ren, "Secure ranked keyword search over encrypted cloud data," in *IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10)*, pp. 253-262, 2010.
[27] C. Wang, N. Cao, K. Ren and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467-1479, 2012.
[28] C. Wang, S. Chow, Q. Wang, K. Ren and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362-375, 2013.
[29] W. Zhang, S. Xiao, Y. Lin, T. Zhou and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1566-1577, 2016.

Biography

Lina Zou was born in 1980. She graduated and received her M.S. degrees from Northeastern University in 2007. Lina Zou is a lecturer in the department of computer and mathematics, Shenyang Normal University, Shenyang, China. She has published many international papers indexed by EI and SCI. Her research interests include the theory of computer, Mathematics, Information Security and Intelligence Algorithm.

Xueying Wang was born in 1966. She is a professor in department of computer and mathematics and Software College, Shenyang Normal University. She received her B.S. and M.S. degrees from Wuhan University. Her research interests include Enterprise informatization and Innovation and entrepreneurship education.

Shoulin Yin received the B.Eng. And M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2013 and 2015 respectively. His research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. He received School Class Scholarship in 2015. Email: 352720214@qq.com.