SECURE RANK ORDERING SEARCH SCHEME ON OUTSOURCED ENCRYPTED CLOUD DATA USING MULTIPLE KEYWORDS

[1]R.Bhavithra, [2]A.Nithyasri, [3]S.Pavithra.

[1]PG Scholar, IT Department Vivekanandha College of Engineering for Women (Autonomous),Thiruchengodu,India

[2,3]Assistant Proffessor, IT Department Vivekanandha College of Engineering for Women (Autonomous) Thiruchengodu, India

[1]bhavithraravi@gmail.com, [2]nithi.becse@gmail.com, [3]pavitrasaran92@gmail.com

Abstract: With the character of low maintenance, cloud computing enables data owners to outsource their data to the cloud server and allows data users to retrieve those data from the cloud server. Usually sensitive data are encrypted by data owners before uploaded to the cloud for protecting the privacy of the data, which makes difficult to search the encrypted cloud data. Secure rank ordered search scheme has been presented in order to overcome this issue. The proposed scheme includes TF×IDF model, vector space model, and tree based index structure and kNN algorithm for generating the index, encrypting the index, generating the trapdoor and for searching the encrypted data. In order to bring security against revoked users, the proposed scheme supports data user revocation.

Index Terms: cloud computing, keyword search, ranked search, revocation, searchable encryption, TF×IDF.

1. INTRODUCTION

The main goal of cloud computing is to provide on-demand outsourcing data services for the data users. With the advent of low cost and high performance, cloud computing enables data owners to outsource their data to the cloud server and allows data users to retrieve those data from the cloud server. However data users and data owners may not fully trust the cloud server. Because the cloud server that has the user's data may access the user's sensitive information (such as e-mails, personal health records, photo albums, company finance data, tax documents etc.,).

In order to address the privacy requirements, sensitive data are usually encrypted before uploaded to the cloud servers. Simply encrypting the data is not always enough to ensure privacy. The adversary can still learn sensitive information through observing access patterns made by data users to the cloud storage. Apart from tracking the access patterns by the data user, encrypting the sensitive data limits the usability of data due to the difficulty of searching the encrypted data over the cloud.

On addressing the above problem, several solutions have been designed by researchers. Their solutions for the issues associated with encrypted data in the cloud are fully homomorphic encryption (FHE) and oblivious RAMs. Fully homomorphic encryption (FHE) supports arbitrary computation on cipher texts. It enables the program construction for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result. However, this method works only for medium sized datasets and it is too impractical to implement on a large datasets. Oblivious RAM completely hides the access pattern of the encrypted data in the cloud by continuously shuffling the memory. Both fully homomorphic encryption (FHE) and oblivious RAM are impractical due to their computational overhead. So various search schemes have been designed by researchers to achieve search functionality over the encrypted cloud data. Typically those searchable encryption schemes works as follows: First, the data owner generates the keywords from the unencrypted document. Then the data owner sends the encrypted form of those keywords and documents to cloud server. Second, when the data user needs to retrieve the encrypted document over the cloud, he should query the keyword. So that he send the cipher text of the relevant keywords according to the document that is needed. Finally, the cloud server matches the cipher text of the keywords sent by the data user with the cipher text of the keywords sent by the

data owner and provides the relevant search results to the data user.

Even though various searchable encryption schemes have been developed, it is still difficult to achieve search functionality over encrypted document as flexible as search functionality over unencrypted document. Users can search the document as their wish in the case of unencrypted document, but not in the case of encrypted document. The main contributions of this paper are listed as follows: First, we provide a dynamic multi-keyword ranked search scheme over an encrypted cloud data. Second, we provide a revocation scheme that support user revocation through a novel revocation list without updating the secret keys of the remaining users.

The remainder of this paper is organized as follows: Review of the related works is discussed in Section II, and Section III outline the system model, threat model and design goals. The proposed schemes are presented in the Section IV. Performance analysis is presented in Section V. Section VI concludes the paper.

2. RELATED WORK

Searchable encryption schemes motivate the data owners to outsource their data to cloud and allow the users to make a keyword search to retrieve the encrypted data. Searchable encryption scheme include single keyword search scheme, multi-keyword search scheme, single keyword ranked search scheme, multi-keyword ranked search scheme, dynamic searchable encryption scheme. Among them multi-keyword ranked search scheme provides better results than other searchable encryption schemes. There are two types of searchable encryption schemes. They are Searchable Public Encryption (SPE) and Searchable Symmetric Encryption (SSE).

2.1 Searchable Public Encryption

Boneh et al. [3], proposed the first searchable public encryption, however this scheme is impractical due to computational overhead. Boneh et al. [4], proposed the conjunctive, subset, and range queries over the encrypted data. Hwang et al. [5], proposed a conjunctive keyword search scheme that supports multi-keyword search. Zhang et al. [6], proposed an efficient public key encryption with conjunctive-subset keyword search. Conjunctive keyword search schemes only retrieve the documents that contain all of the keywords in the query. Disjunctive keyword search schemes retrieve the documents even if any of the single keyword in the query contained in the document. Predicate search schemes support both conjunctive keyword search scheme and disjunctive keyword search scheme. However, these search schemes cannot retrieve the acceptable ranked results. Yu et al. [7], proposed a multi-keyword top-k retrieval scheme with fully homomorphic encryption (FHE). Qin et al. [8], proposed a ranked query search scheme that uses a mask matrix to retrieve the ranked results. In general, Even though searchable public encryption (SPE) allows search functionalities over the encrypted data, it is still less efficient than the searchable symmetric encryption (SSE).

2.2. Searchable Symmetric Encryption

Song et al. [10], proposed the first searchable symmetric encryption. Goh [11], proposed a formal security definitions for searchable symmetric encryption schemes. Curtmola et al. [12], proposed two schemes named as symmetric searchable encryption 1(SSE1) and symmetric searchable encryption 2(SSE2) to provide security against chosen-keyword attack (CKA1) and adaptive chosen-keyword attack (CKA2) respectively. However, these early works are simple in terms of functionality due to the only usage of single keyword.

Multi-keyword Boolean search schemes allow users to query multiple keywords. However these search schemes cannot retrieve acceptable ranked results.

Ranked search scheme retrieves the most relevant documents according to the keyword. Single keyword ranked search schemes [13], [14] have realized ranked search using order preserving techniques. However, they are simple in terms of functionality. Cao et al. [15], proposed privacy-preserving multi-keyword ranked search scheme with the help of “coordinate matching” technique. However their scheme does not consider the importance of different keywords. Sun et al. [16], proposed privacy-preserving multi-keyword text search in cloud supporting similarity-based ranking with the help of searchable index tree based on vector space model. However, this search scheme results in precision loss. Zhang et al. [17], [18] proposed secure ranked multi-keyword search for multiple data owners in cloud

computing. In this scheme relevant documents are retrieved with the help of "Additive order preserving function". However these schemes cannot support dynamic operations.

Various dynamic searchable encryption schemes have been designed by researchers to support dynamic insertion and deletion of the documents. Cash et al. [19], proposed a data structure for keyword/identity tuple named "T-sets" and then they proposed a dynamic searchable encryption scheme [20], based on the T-set structure. But this scheme doesn't realize multi-keyword ranked search functionality. Zhihua et al. [21], proposed a secure and dynamic multi-keyword ranked search scheme. This scheme realized ranked results and dynamic operations using tree based index structure and combination of vector space model and widely used TF×IDF model. However, their scheme doesn't consider the user revocation when it comes to multi-user scheme.

III.System Model, Threat Model And Design Goals

3.1.System Model

Let us consider a cloud computing architecture by combining with an example that a company uses a cloud to enable its staffs to access document. The system model consists of three different entities: Data owner (i.e., the company manager), Cloud server, Data users (i.e., the staffs) as illustrated in Fig: 1.

- Data owner. The data owner takes charge of user registration, searchable index creation, key generation, searchable index encryption and user revocation. Then the data owner outsources revocation list, encrypted documents and encrypted indexes to the cloud. The data owner then distributes the keys to the data users for generating the trapdoor and for decrypting the documents in a secure way. In the given example, the data owner acted as an administrator of the company. Therefore, we assume that data owner is trusted by other parties.
- Cloud server. The cloud server in this system model is considered as "honest-but-curious". It is an intermediate entity which stores the revocation list, encrypted documents and encrypted indexes sent by the data owner. When a data user sends the trapdoor for retrieving the document, the cloud server checks whether the user is authorized user or not through the revocation list. Then it calculates the relevance score based on the trapdoor that has sent by the data user and the index that has sent by the data owner. Cloud server then executes the search over the index tree. Finally, the cloud server returns the relevant encrypted documents to the data user. Besides, upon receiving the update information from the data owner, the cloud server needs to update index, encrypted document collection, revocation list according to the information received from the data owner.

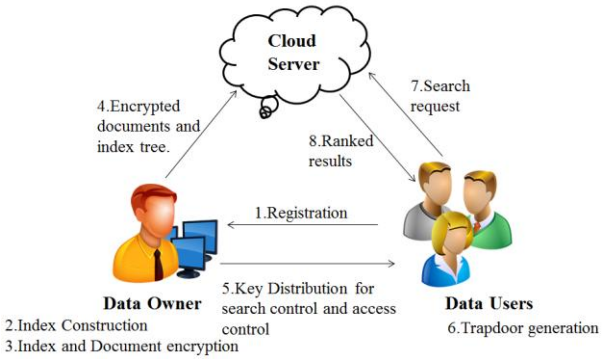

Figure1: System Model

- Data users. Data users task include user registration to the data owner, receiving keys from the data owner for generating trapdoor, retrieving the encrypted document from the cloud server using the trapdoor and at last decrypting the document using the key provided by the data owner. In our example data users play the role of staffs of the company. Note that, staff resignation and the new employee participation is possible frequently. Therefore, the revocation list is dynamically changed.
- Data users. Data users task include user registration to the data owner, receiving keys from the data owner for generating trapdoor, retrieving the encrypted document from the cloud server using the trapdoor and at last decrypting the document using the key provided by the data owner. In our example data users play the role of staffs of the company. Note that, staff resignation and the new employee participation is possible frequently. Therefore, the revocation list is dynamically changed.

3.2. Threat Model

In our threat model, both the data owner and authorized data users are considered as trusted. But the cloud server is considered as "honest-but-curious" (i.e., semi trusted cloud server). The cloud server may predict and analyses the encrypted documents based on the information provided by the data owners and data users. We consider two threat models based on possible information gathered by the cloud server.

- Known cipher text model. The cloud server has encrypted documents and corresponding indexes, which are sent by the data owner. Using those information, cloud server can conduct cipher-text only attack.
- Known background model. Upon receiving the trapdoor from the data user, the cloud server can predict and analyse the relationship between trapdoor and indexes. That is to say, the cloud server has the ability to know about the index construction.

3.3 Design Goals

Based on the threat model, our system has the following design goals

- Data confidentiality. The encrypted document that is outsourced to the cloud server should be identifiable only by data owner and authorized users.
- Trapdoor uniqueness. The cloud server should not be able to determine trapdoors of the documents when a data user searches the same document stored at the cloud server again and again.
- Privacy protection of index, trapdoor and revocation list: The cloud server should not be able to identify the contents of the index, trapdoor and the revocation list.

4. PROPOSED SCHEME

This section describes the user registration, user revocation, searchable index creation, key generation, searchable index encryption, file upload, key distribution, trapdoor generation, file searching, relevance score calculation, search process and dynamic update operation.

4.1. User Registration

When a data user Xi makes a registration, the data owner computes the unique user ID (U_{id}). The data owner then adds the user ID (U_{id}) in the revocation list.

4.2. User Revocation

User revocation is performed by the data owner to ensure the confidentiality against the revoked users. Data owner update the revocation list each and every day even if no data user has been revoked on the day. Revocation list is bounded by unique ID (Uid) of the data user and updated timestamps. Finally, the data owner outsources the revocation list to the cloud server.

4.3 Searchable Index Creation

In our scheme, the data owner constructs the index based on the keyword balanced binary (KBB) tree structure. "Term frequency (TF) × inverse document frequency (IDF)" model are used to rank the keyword.

- The data owner generates the keywords from the unencrypted document (i.e., Multi-keyword generation) as shown in Fig: 2.
- TF × IDF model is used to rank the document according to the keywords. Term frequency (TF) means the number of times a generated keyword appears within a document. Inverse document frequency (IDF) is a statistical weight used for measuring the importance of a keyword in the document collection. IDF value of keyword in a single document is calculated by computing log value of division of cardinality of the document collection by number of documents containing that keyword. Finally TF× IDF value of the keyword in document is calculated by multiplying TF value and IDF value of that keyword as illustrated in Fig: 2.
- The data owner construct the index based on the keyword balanced binary tree structure using the normalized TF×IDF values. Fig 3 shows how the keyword balanced binary tree is constructed. First, Leaf nodes are constructed from the documents. Then the internal nodes are constructed based on the leaf nodes i.e., index value of 2 documents are compared in order to construct the internal nodes. After the index construction, the data owner encrypts the index.

DOCUMENT 1			
Term	TF	IDF	TF×IDF
Privacy	4	0.2	0.8
Preserving	6	0.2	1.2
Survey	5	0.5	2.5

DOCUMENT 2			
Term	TF	IDF	TF×IDF
Privacy	6	0.2	1.2
Preserving	7	0.2	1.4
Search	4	0.2	0.8

DOCUMENT 3			
Term	TF	IDF	TF×IDF
Multi	5	0.5	2.5
Keyword	4	0.5	2
Search	2	0.2	0.4

Figure 2: Example for TF×IDF model

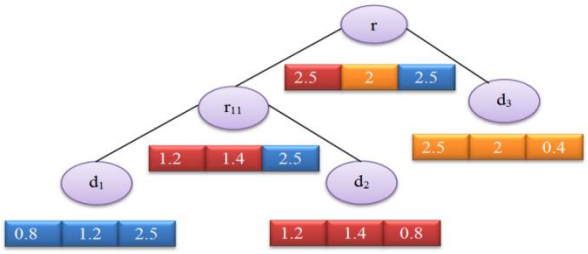

Figure 3: Example for Keyword Balanced Binary Tree

4.4. Key Generation

In our proposed scheme, Multi-keyword ranked search scheme is achieved through kNN computation scheme. The data owner takes charge of key generation as follows:

The data owner randomly generates the two (n×n) invertible matrices such as M_A and M_B, and randomly generated n bit vector N where n is the number of keywords in the single query as illustrated in Fig: 4. Note that if four keywords are taken from each document, then 4 bit vector and 4×4 invertible matrices should be generated randomly as a key.

N=001 (i.e., 3 bit vector)

M_A = $\begin{pmatrix} 7 & 2 & 1 \\ 0 & 3 & -1 \\ -3 & 4 & -2 \end{pmatrix}$ M_B = $\begin{pmatrix} -2 & 3 & -9 \\ 8 & -11 & -34 \\ -5 & -7 & 21 \end{pmatrix}$

Figure 4: Example for key generation

4.5 Searchable Index Encryption

Keyword balanced binary tree is encrypted using the key generated by the data owner. Index vectors are split into two random values based on the n bit vector N as illustrated in Fig: 5.

I_v (i.e., Index Value) →(I_v', I_v'')

If N[i] = 0, I_v'[i] and I_v''[i] will be set equal to I_v[i]; If N[i] = 1, I_v'[i] and I_v''[i] will be set as two random values whose sum equals Iv[i]. Finally, EI_v (i.e., Encrypted Index Value) = $M_A^T I_v$', $M_B^T I_V$'' as illustrated in Fig: 6.

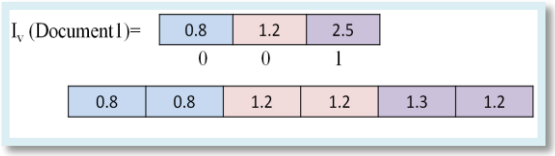

Figure 5: Example for Index Splitting

Figure 6: Example for Encrypted Index Value

4.6 File Upload

Data owner upload the encrypted document collection and encrypted index securely to the cloud server.

4.7 Key Distribution

Data owner distributes the secret key set N, M_A, W_q Where W_q – Keyword set with IDF values and key information of document decryption to the authorized data users.

4.8 Trapdoor Generation

The data user generates the keyword set (K) for searching. The unencrypted query vector (Q) is generated from the keyword set (K) by applying the key that has sent by the data owner, with the length of n, where n is the number of keywords in the query. Query vector is split into two random vectors based on the n bit vector N.

Q →(Q', Q'')

The data user generates the trapdoor by multiplying the invertible matrices with Q' and Q'' respectively.

Trapdoor $\longrightarrow$ $\{M_A^{-1}Q`, M_B^{-1Q``}\}$

Finally, the data user searches the encrypted documents using the generated trapdoor.

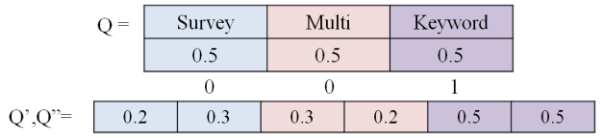

Fig 7: Example for Query vector splitting

Figure 8: Example for Trapdoor Generation

4.9 File Searching

When a data user needs to search the document over the cloud, he will then send the trapdoor of the document and the number of documents he wants to retrieve to the cloud server.

4.10 Search Process

Upon receiving the trapdoor from the data user and the number of documents he wants to retrieve, cloud server calculates the relevance score using the trapdoor from the data user and the normalized TF×IDF values in the nodes of the indexes. Finally, the cloud server returns the top ranked results based on the relevance scores of the document as illustrated in the Fig: 9.

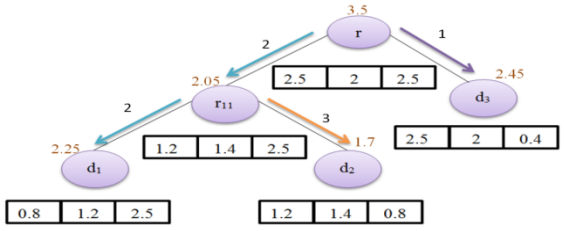

Figure 9: Example for Searching Process

4.11 Dynamic Update Operation

Data owner may need to update the document after outsourcing the documents to the cloud server. Dynamic update operations include insertion and deletion of the document. The data owner has the unencrypted indexes and documents.

If the data owner needs to insert the document, he will then send the encrypted document along with corresponding index to the cloud server. Since the searchable encryption scheme has been designed as the tree based index structure, the data owner sends the new node that contains the encrypted document and corresponding indexes. The new node will be added to the leaf node of the subtree. In case if there is any fake node to balance the tree before inserting the new node, the data owner replaces the fake node with new node.

If the data owner needs to delete the document, then the data owner deletes the node that contains the document id to be deleted. If the tree is not balanced because of the deleted node, then the fake node would be created which contains the document ID as null to balance the tree.

5. PERFORMANCE ANALYSIS

We implement the proposed scheme using ASP.NET language in windows 8 operating system. The efficiency of user revocation, index construction, trapdoor generation, search process, dynamic update operation are obtained with an Intel ® core ™ i5-337U CPU @1.80 GHz.

5.1 User Revocation

O (m) users are checked in the user revocation. So the time complexity for user revocation is O (m).

5.2 Index tree construction

Index tree construction involves generating O (m) nodes and two multiplications of (n × n) matrix. As a whole, the time complexity for index tree construction is O (n^2m). The space complexity of index tree construction is O (nm).

5.3 Trapdoor Generation

Trapdoor generation involves two multiplications of a (n × n) matrix, thus the time complexity is O (n^2).

5.4 Search Process

Search process involves checking the relevance scores. The time complexity of relevance score calculation is O (n). Thus, the time complexity of search is O (Θ n log m).

5.5 Dynamic Update Operation

Dynamic update operation involves updating leaf node and encrypting indexes. The data owner needs to update log m nodes and encrypting the indexes takes O (n^2) time. Thus, the time complexity for update operation is O (n^2 log m).

6. CONCLUSION

In this paper, secure rank ordering search scheme is proposed. This scheme supports multi-keyword search which provides ranked acceptable results. This scheme also supports dynamic operations such as dynamic insertion and deletion. The revocation scheme in this paper provides security against revoked users. Keyword balanced binary tree is constructed for building index. Combination of vector space model and widely used TF×IDF model is used to deal with the multi-keyword ranked search scheme. Revocation list is constructed to provide security in the case of multi-user scheme.

REFERENCES

[1] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[2] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM (JACM),vol. 43, no. 3,pp. 431–473, 1996.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology—Eurocrypt. Springer, 2004, pp. 506–522.

[4] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Theory of cryptography, Springer, 2007, pp. 535–554.

[5] Y. Hwang and P. Lee, "public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proc.Pairing. Springer, 2007, pp. 2–22.

[6] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," J. Netw. Comput. Appl.,vol. 34, no. 1, pp. 262–267, 2011.

[7] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Towards secure multikeyword top-k retrieval over encrypted cloud data," IEEE Trans.Dependable Secure Comput., vol. 10, no. 4, pp. 239–250, Jun. 2013.

[8] Q. Liu, C. C. Tan, J. Wu, and G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments," in Proc. IEEE INFOCOM, 2012, pp. 2581–2585.

[9] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li,"Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 11, pp. 3025–3035, Nov. 2014.

[10] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55.

[11] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrintArchive,vol. 2003, p. 216, 2003.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88.

[13] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu, and D.W. Oard, "Confidentiality-preserving rank-ordered search," in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM, 2007, pp. 7–12.

[14] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+ r: Topk retrieval from a confidential index," in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 439–449.

[15] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in IEEE INFOCOM, April 2011, pp. 829–837.

[16] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 71–82.

[17] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing," in Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on. IEEE, 2014, pp. 276–286.

[18] R.S.Archana Vishveswari and P.Selvi, "Secure Dependable Data Storage in Cloud Computing", International Journal of Innovations in Scientific and Engineering Research(IJISER), vol. 1, no. 108, 1-5, 2014.

[19] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Ros¸u, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in Advances in Cryptology–CRYPTO 2013. Springer, 2013, pp. 353–373.

[20] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. of NDSS, vol. 14, 2014.

[21] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang," A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data" in IEEE Transactions on Parallel and Distributed Systems, vol.27, 2016.