Secure and Efficient Search Over Cloud: A Comparative Study of various Searching Techniques

Rajpreet kaur
Research Scholar, CGC, Landran, Mohali, Punjab, India.

Manish Mahajan
Head of Department, Computer Science & Engineering, CGC, Landran, Mohali, Punjab, India.

Abstract – In recent information technology, cloud computing is a dominant trend. Due to large number of data users and large storage of data over cloud servers, it becomes very important to ensure efficient retrieval over cloud. For effective search of cloud data security and result ranking is very necessary. Paper first, describes various search techniques briefly. After that some problems related to privacy and efficient retrieval are illustrated. Then an evaluation is made, which shows how TRSE is beneficial over other techniques. TRSE (Two round searchable encryption) is a secure search scheme, which uses vector space model and homomorphic encryption.

Index Terms – Cloud, Efficient search, Security, ranking, Homomorphic Encryption, Vector Space Model.

1. INTRODUCTION

Cloud computing is a software through which one can take hardware and software resources on rent. Cloud computing is the delivery of computing services over the Internet. In cloud computing, resources like storage, networking facilities, application services etc are provided to user without actual installation and management of hardware and other system. Hence, it is not incorrect to say that all the resources are stored over cloud which is a large shared pool of computing resources.

As Figure1 elaborates, cloud computing provides various types of benefits in field of computing. Cloud services are flexible, because one can use as much resources as he require by paying for them. All the data stored at cloud servers is automatically updated by software programs. As the data is centralized at cloud servers, in case of PC failures data can be recovered easily. The most important thing in computer environment is the safety of data. Only an authorized person can access the files stored at cloud. Only the data owner and data users with the access rights possess the secret keys related to encryption and decryption.

As the demand of cloud services is increasing, more and more sensitive data and confidential information is being stored over cloud servers. Emails, personal health records, private videos and photos, financial data, government documents, military information etc are stored on cloud servers. So, data encryption becomes very necessary for protecting data privacy and for preventing unauthorized access to private data. So, for security reasons cloud data is encrypted before outsourcing for end–to-end data confidentiality.

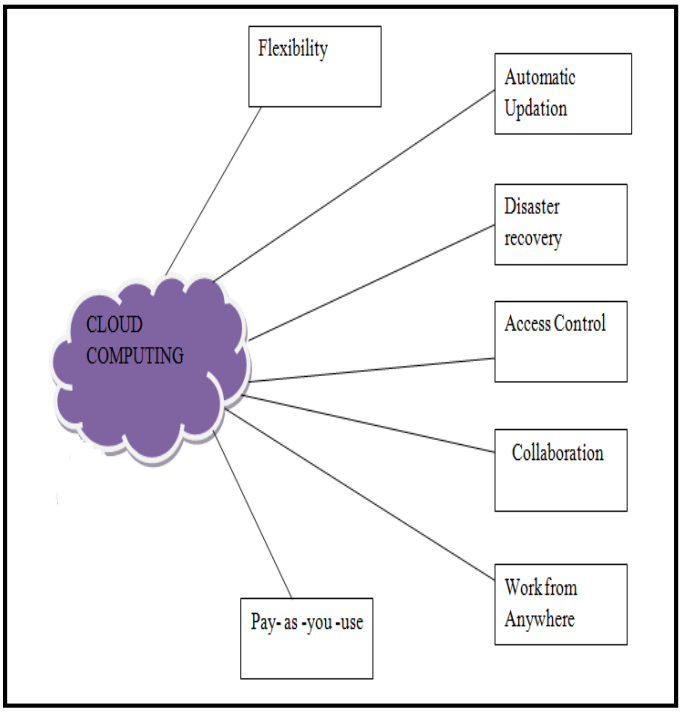

Figure1. Benefits of cloud computing

However, efficient search of encrypted data becomes very difficult and challenging task, because there could be large amount of outsourced data files. In addition to this, on cloud servers a large amount of data is outsourced for a large numbers of users. But during a particular session a user may be interested in retrieving a specific file of his interest only. It can be done using keyword based search method. Keyword based search methodology is very popular in plaintext search scenarios; in which user can selectively search files of his interest using the keyword. But unluckily, when data is in encrypted form user cannot perform keyword based search queries. Further, keyword privacy is also required at cloud server for encrypted data files. Traditional search techniques allow user to securely search encrypted data through keyword

based queries. Yet, search results are based only on Boolean keyword search methods and not capture any relevance of result files with the given keyword.

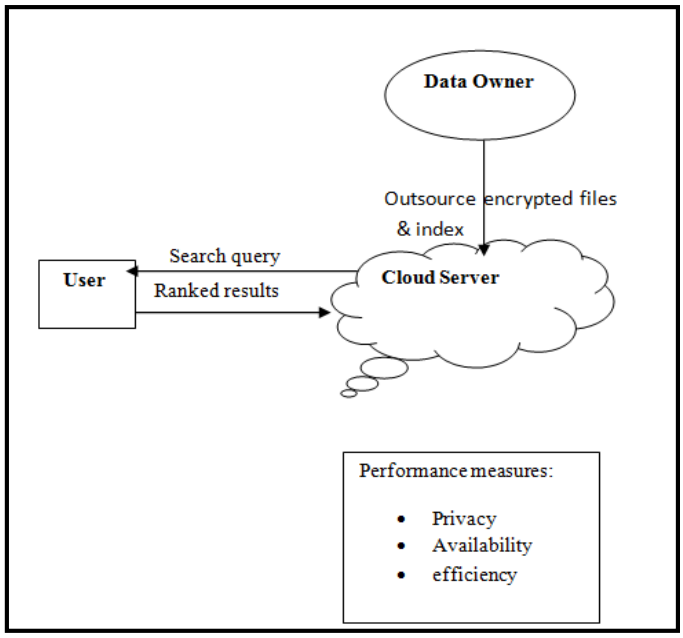

Figure2. Searching over cloud

These previous methods suffer from two kinds of limitations. Firstly, user is a novel person without any knowledge of encrypted data and storage mechanisms used. So, user generally goes through all result files in order to get files of his own need. Large processing overhead occurs for this. Due to this, method is not such efficient. Another limitation is the network traffic created by generating so many result files. In short, the main drawback of traditional schemes is that they lack to ensure effective results and file accuracy. Still, Information retrieval (IR) community has utilizing various mechanisms for effective data searching. Different concepts are being used by these mechanisms like relevance scoring and ranking the results orderly.

The Figure 2, which is given above illustrates a simple model of ranked search over cloud. Three important things on which performance of search mechanism depends are privacy, availability (accessing data in appropriate time), and efficiency.

2. A LITERATURE STUDY OF VARIOUS SEARCH SCHEMES

A top-down methodology is described below in figure 3, which gives a stepwise construction of a secure search scheme. This methodology starts with search function and then decomposed to basic steps of information retrieval and other primary operations. Further, it is decomposed to cryptographic methods for encryption.

After it a proper encryption scheme is found to encrypt the data while simultaneously allowing data operations. Although index structure makes retrieval of information much more efficient, yet information needs to be handled carefully to prevent leakage to cloud servers. The reason behind this is that server has background information about query statics. As a result, sometimes the encryption primitive itself may need to be adapted to meet privacy requirements.

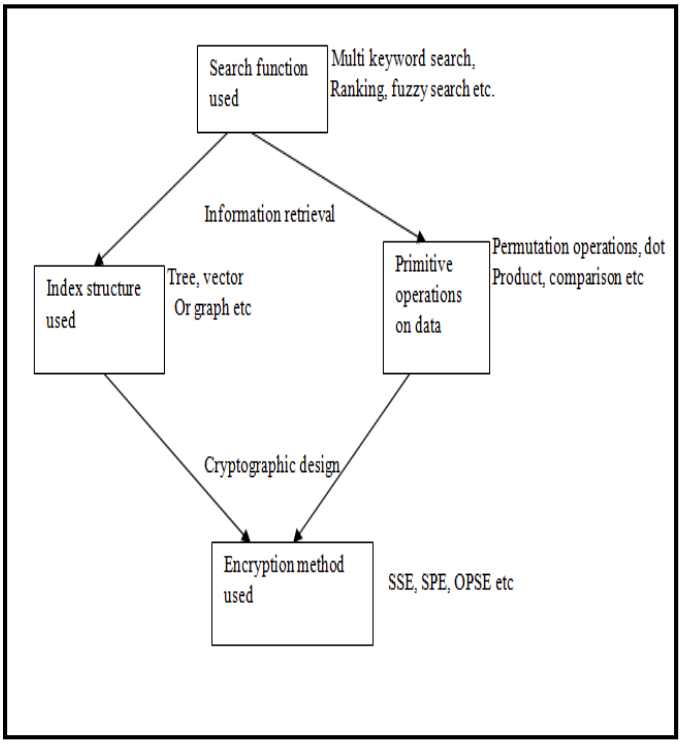

Figure3. Top-down methodology for secure search scheme

Further some techniques for searching are discussed based on different factors:

2.1 Symmetric Searchable Encryption

Secure searchable encryption (SSE) is a deterministic encryption scheme generally based on symmetric key of cloud owner and data user. Curtmola *et al.* proposed SSE [8], which uses inverted index scheme and also some basics of permutation and pseudorandom functions. This method of searching is quite efficient. Roughly speaking, the index consists of blinded keywords $f_k(w_i)$ *a*nd lists of FIDs containing w_i keywords, where f () is a pseudorandom function and k is the secret key. The search trapdoor is also in the same form so that the server can perform matching. However, it only supports single-keyword exact query.

2.2 Scalar-Product-Preserving Encryption

The basic computing function used in SPE is dot product of vectors. An SPE scheme [19] preserves the dot product

between two *d*-dimensional vectors (e.g., a query vector $\vec{q}$ and a database record $\overrightarrow{pi}$). In simple words, the secret key contains one (d + 1) -bit vector $\vec{s}$ and two (d + 1)×(d + 1) invertible matrices$\{M_1, M_2\}$. Every data vector $\overrightarrow{pi}$ and query vector $\vec{q}$ are extended, $\overrightarrow{pi}$ is randomly scaled, and both of them are split into two random vectors $\{\overrightarrow{p_i}', \overrightarrow{p_i}''\}$ and $\{\vec{q}', \vec{q}''\}$ respectively. They are encrypted as $\{M_1^T\overrightarrow{p_i}', M_2^T\overrightarrow{p_i}''\}$ and $\{M^1{}_{-1}\vec{q}', M^2{}_{-1}\vec{q}''\}$ respectively. The server can then recover $r\vec{q}\cdot\overrightarrow{pi}$ from ciphertexts, without knowing the original vectors. For each dimension, the method of splitting is controlled by the corresponding bit in key vector $\vec{S}$, which is proven to provide sufficient security against known ciphertext attacks [19]. But it creates a lots of computational work.

2.3 Order-Preserving Symmetric Encryption

Order preserving symmetric encryption is a method which improves efficiency by preserving order of search results. In OPSE [2], the numerical ordering of plaintext is preserved after encryption. Boldyreva *et al.* [2] provide the first cryptographic construction of OPSE. It works in the security frameworks like pseudo random permutations and functions. It can be regarded as a function g (.), from a domain D = {1, …, *M*} to a range R={1,…,*N*}.

2.4 Ranked search over encrypted data

Ranking of search results is very important for efficient utilization of search methodology. Inverted index structures are very useful for ranked keyword search. At first, when user generates a query using search keyword, relationship between terms is identified using index values of terms. After that operations (like matching or sorting etc) are performed on retrieved data to generate ranks. For example a mathematical function is given for calculating relevance score of words and files to calculate ranks:

$$Score(t, F_d) = \frac{1}{|F_d|}\cdot\left(1 + \ln f_{d,t}\right) \qquad (1)$$

Where t is the search keyword (term), $f_{d,t}$ denotes the term frequency (TF) of keyword t in fileF_d, and $|F_d|$ is the number of indexed keywords in$\mathrm{F_d}$.

Zerr et al. proposed a method [20] for secure ranked search over encrypted data. Their approach was based on posting list (unique index values for each term). The approach was really efficient, but re-identifying the keywords for each mapping requires much processing. Wang [23] presented a scheme named RSSE (Ranked symmetric searchable encryption), which was basically derived from existing SSE [8].

After that OPSE [2] was proposed by some authors, which is more practical in improving performance. This approach ensures the order of plain text in the encrypted form. Main problem tackled by this method is the information leakage at cloud because of background knowledge possessed by server. Then Xia et al. [17] proposed one-to-many (OPM) mapping. Multiple keyword ranking was still not considered to such extent. Cao gave MRSE (Multi-keyword ranked search) [21]. Some other schemes secure inner-product mechanism that was used by SPE to securely search in k-nearest neighbor method [24].

2.5 Fuzzy logic based search

Fuzzy logic can be said a similarity based search. In this a fuzzy set is created by taking similar terms to a given query keyword. Similarity can be defined by using edit distance between various words. For example, the edit distance between "Britney" and "briny" is 2. When the edit distance equals 1, it is usually called fuzzy search. Thus, in a similarity search problem, given a keyword w and an edit distance d, a search execution should return a set of files $\{FID_{w_i}\}$, where $d(w, w_i) \le d$.

A straightforward method is to build a similarity keyword set that incorporates not only the exact keywords but also those that differ by up to edit distance d. Also, the search would require a trapdoor for each similar word to the query keyword. However, this incurs high storage and computation overhead at the server side due to the large number of possible similar keywords. Thus, in [25], Li et al. proposed using two data representation techniques to enhance the efficiency for fuzzy search.

2.6 Two Round Searchable Encryption (TRSE)

In existing search schemes based on OPE (Order preserving Encryption) ranking is done on server-side to improve the efficiency of retrieval process. However, by doing this privacy of encrypted data is violated. This violation to security is undesired in applications which are security-oriented like, personal health records, military information etc. So, the tradeoff between privacy and efficiency is not tolerable. More security guarantees can be ensured if ranking is done on user side. But, by doing so, a large computational load and communication overhead appears at user side. This is due to communication and interaction between server and user for score calculations and index generations.

Thus, user-side ranking helps in improving privacy. But this occurs at the cost of extra computational overhead. So, practical use of this is very difficult. Now, we come to the study of a new encryption scheme, which uses the novel technologies of IR community and cryptography. This approach includes homomorphic encryption and a vector space model. In this technique, data owner encrypts the data using homomorphic encryption. When cloud server receives a query based on multi-keywords, it calculates the relevance score from encrypted index from index structures stored at cloud. Then, it returns the scores in encrypted form to the user. Next, user decrypts the scores and gets the identifiers of top-k highest scoring files.

Now, user requests to server for top-k files by sending identifiers generated. Hence, the retrieval is a two-round process between cloud and user. This scheme is mentioned as TRSE (two-round searchable scheme) scheme. Here score calculation done at server side, but ranking is done by user. Two main models used in TRSE are described below:

2.6.1 Vector Space Model

While $tf - idf$ scoring scheme is used to calculate weights for single keyword search, vector space model generate scores a file on multikeyword. In vector space model [35] algebraic representation is used, which considers file as a vector. Every dimension of vector represents a term in file. The value of term is non-zero in file if it occurs, otherwise value becomes zero. Vector space model supports multi-keyword and non-binary representation. It allows calculation of similarity with a continuous degree. It enables multi-keyword top-k retrieval. A query is represented by vector r $\vec{q}$, where each dimension in the vector is 0 or 1 according to presence or absence. The relevance score of file f for a given query $q(Score_{f,q})$ is calculated using inner product of the two vectors. A formula is given as: $Score_{f,q} = \overrightarrow{v_f} \cdot \vec{q}$. After computing the scores files can be ranked in ascending or descending order. Thus, top most relevant files can be searched.

2.6.2 Homomorphic Encryption

To release user from computational burden, computational work can be done at server side. For this an encryption scheme is desired which can operate at server side but also guarantees the security. This can be achieved through homomorphic encryption .Because it allows specific operations to be applied on ciphertext. Result is same as the result of operation performed on plaintext of this encrypted data. But, result is also in the form of ciphertext. Thus, homomorphic encryption can perform operation on ciphertext without knowing anything about its plaintext. Fully homomorphic encryption [34], is inefficient for practical use. Luckily, a vector space model is used. So, only a little number of operations like multiplication and addition of integers are required to calculate relevance scores. So fully homomorphic encryption can be simplified to an approach which supports only integer calculations. This improved version of homomorphic is called fully homomorphic encryption over the integers (FHEI) [37].

3. PRIVACY RELATED ISSUES

Cloud computing [28], provided data users with easy and more efficient ways to outsource data. With more and more advancements in IT technology, more and more private data is being outsourced over cloud. There are many controversies relating to privacy of these private data files. Reports on data loss and attacks on privacy occurs time to time [29], [30] in cloud computing systems. Cloud itself is the main threat for data privacy [42]. The reason of it is that cloud service providers are able to control and monitor data and the communication that occurs between different users. Hence cloud service providers can affect the data files with will, lawfully or illegally.

Users encrypt their personal data before outsourcing to ensure security and confidentiality. But with encryption data utilization becomes very difficult task. Although encrypted data utilization is possible with certain mechanisms yet cloud works as an intermediate here. Due to this there are chances of leakage and deteriorating confidentiality of files over cloud.

Further discussing, data owners may share the outsourced data with a large number of users. But users might want to retrieve the files of their interest only. Users always seek the data files which are relevant to their need. Most popular way of searching cloud data is through keyword-based retrieval. But by using this technique user generate queries based on the keywords and get all the files merely on the basis of presence or absence of that keyword. This technique does not bother about the relevance of files with search requirements. To improve these shortcomings, relevance score of files can be helpful. We can calculate the relevance of files with the keyword with mathematical formulas and generate the list of ranked results based on relevance. In this way search results become more efficient.

Various schemes for searchable symmetric encryption (SSE) have been designed for enabling search over encrypted data. Traditional schemes [12], [3] enables user to securely search the ciphertext, but that schemes are based on Boolean search, i.e., only existence of keyword in file. They do not consider any relevance of search keyword with these retrieved results. To improve efficiency and security of search top-k single keyword retrieval technique is used in [16], [20], [31]. With more enhancements authors in [5], [32] attempt to solve issues of top-k multi-keyword search over ciphertext. These methods suffer from the problems like Boolean representation and tradeoff between efficiency and security. In these approaches files are ranked only based on the number of retrieved keywords, but do not consider any semantic relationship of files with search keyword. Tradeoff between efficiency and security is also not required in privacy-assured search mechanisms.

Information leakage over server can be avoided if all the work is done by user and preventing clod to involve in ranking. However, the user has a limited computational power and a high computational overhead occurs from which efficiency suffers. The main issue in case of multi keyword –based information retrieval is that how can one effectively utilize the search results without sacrificing security of information.

3.1 Shortcomings of SSE and other search techniques

3.1.1 Information Leakage- The main threat to privacy is cloud itself. The cloud server considered is based on "honest-but-

curious" [16] model. Which means the server will honestly work in accordance with the predefined constraints. But server is also curious to analyze the stored data to get more and more information.

Although all the documents, indices and queries send to cloud are in encrypted format. But server can still get related information using statics. This possible leakage of information is referred as statistical leakage. This leakage can be of two possible types, regarding term distribution or interdistribution. Term distribution of term t can be defined as t's frequency distribution of relevance scores on each file $i(i \in C)$. Similarly, interdistribution of a file f is defined as $f's$ frequency distribution of scores of each term $j(j \in f)$. Term distribution and interdistribution have particular value [20]. These values can be calculated directly from the ciphertext or by using statistical analysis over access and search pattern used [8]. By access pattern we mean the way in which keywords and corresponding files have been retrieved. Search pattern refers to whether keywords accessed between two queries are same.

It is being observed that similarity between files can be calculated using distribution information. It is obvious that terms with similar term distribution have simultaneous occurrence. For example, term "Angeles" can commonly occur with "Los" in the any private document related with "Los Angeles". So there term distribution will also be same. Although this document is encrypted but term distribution is being perceived. If someone cracks out the plaintext of "Los", he can easily guess the plaintext for "Angeles".

On the other hand, two files in which interdistribution is same are always of similar type. For example, two medical records from a hospital are of same category. These records must have similar attributes (like title names for different entities can be similar). So this information must be hidden from untrusted cloud server.

3.1.2 Scheme Robustness- Data privacy is endangered by co-occurrence of terms, with a given similarity relevance. Given in [33], co-occurrence of terms means how often two words occur together. It can be measured using various means like t-score or mutual information etc.

4. EVALUATION OF TRSE SCHEME OVER SSE AND OTHER SCHEMES

4.1 Efficiency Improvement

By modifying FHEI TRSE scheme can be simplified than given by Gentry's [34]. But this simplicity costs an increase key size. Compression and modular reduction can be used to decrease the size of ciphertext. Still the key size is very large. This becomes difficult in practical use. As we discussed above, trapdoor is encrypted by user and then sent to the server. With the large key size, encrypted trapdoor might also be of large size. So it creates communication overhead. It becomes very necessary to solve this problem, until we find some new secure search pattern which provides ciphertext of suitable size. Researchers in this area of cryptography community have made many attempts towards practical FHEI which are discussed in [38], [39]. These steps show the signs of further improvements in efficiency of TRSE.

4.2 Security Analysis

From the above discussion, it can be concluded [43] that TRSE provides more security guarantees over SSE schemes. Most of the ranking and computations are done by user himself. Thus, cloud never knows plaintext of files indices and other statistical information. Secondly, cloud also should not learn something about relevance of terms and files. Hence scheme is more robust.

4.3 Enable Update

Practically, in cloud computing systems data updates and modifications were difficult for a searchable encryption technique. But modifications are always essential in real time data storage For example, in medical record of a patient modifications are required frequently. So, for efficient search techniques updating is necessary. For any update both the file and index requires modifications. In TRSE scheme, vector space model uses the $tf - idf$ weighting scheme. In which inverse document frequency (idf) is a factor related to number of files that contain the keyword. This idf factor may change when some term is added or deleted to the file. For avoiding updates to all searchable indices, file vectors should not dependent on each other. Since a searchable index is created for every file, one possible solution to updation problem is to store tf values only in file vectors. Another auxiliary vector is required to store idf values. By doing this, updates are only constrained to auxiliary vector. This can be achieved at the cost that $tf - idf$ needs to be calculated to get relevance score. However, these computations are done at server side, which has large computing power. So, overall efficiency is almost maintained.

5. CONCLUSION

This paper discusses various approaches for secure and efficient retrieval of encrypted cloud data. TRSE ensures more secure and efficient search methodology than other approaches. In future one can work on efficient and fast retrieval approaches which provide relevant results with less computation overheads. Extensive knowledge in the fields of cryptography, information retrieval and database can bring new areas of performance enhancement in this field. Work can be done on performance enhancement by focusing on computational load and also including more and more security factors.

REFERENCES

[1] M. Bellare, A. Boldyreva, A. O'Neill, "Deterministic and efficiently searchable encryption", Advances in Cryptology-CRYPTO , Springer, Berlin/Heidelberg, (2007), pp. 535- 552.

[2] A. Boldreva, N. Chenette, Y. Lee, A. O'neill , "Order-preserving Symmetric encryption", Advances in Cryptology-EUROCRYPT 2009 Springer, Berlin/Heidelberg, (2009), pp. 224-241.

[3] D. Boneh, G. Di, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search", Advances in Cryptology-Eurocrypt, Springer, Berlin/Heidelberg, (2004), pp. 506–522.

[4] J. Bringer, H. Chabanne, "Embedding edit distance to enable private keyword search", Human-centric Comput Inf Sci, vol.2 (1), (2012), pp. 1–12.

[5] N. Cao, C. Wang, M. Li, K.. Ren, W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", Proceedings of IEEE INFOCOM.IEEE, Shanghai, China, (2011) pp 829–837.

[6] Y-C. Chang, M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", Applied Cryptography and Network Security. Springer, Berlin/Heidelberg, (2005), pp 442–455.

[7] M. Chuah, W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data", 31st International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, Minneapolis, Minnesota, USA, (2011), pp 273–281.

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

[9] A. Ibrahim, H. Jin, A. Yassin, D. Zou, "Approximate Keyword-based Search over Encrypted Cloud Data", IEEE Ninth International Conference on e-Business Engineering (ICEBE). IEEE, Hangzhou, China, (2012), pp 238–245.

[10] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou " Fuzzy keyword search over encrypted data in cloud computing" , Proceedings of IEEE INFOCOM.IEEE, San Diego, CA, USA, (2010), pp 1–5.

[11] C. Liu, L. Zhu, L. Li, Y. Tan , "Fuzzy keyword search on encrypted cloud storage data with small index", IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, Beijing, China, (2011), pp 269–273.

[12] DX. Song, D. Wagner, A. Perrig," Practical techniques for searches on encrypted data" , Proceedings of IEEE Symposium on Security and Privacy, IEEE, Berkeley, California, (2000), pp 44–55.

[13] E. Stefanov, C. Papamanthou, E. Shi, " Practical Dynamic Searchable Encryption with Small Leakage", NDSS '14, San Diego, CA, USA, (2014).

[14] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, "Secure ranked keyword search over encrypted cloud data" , 30th IEEE International Conference on Distributed Computing Systems (ICDCS). IEEE, Genoa, Italy, (2010), pp 253–262.

[15] C. Wang, N. Cao, K. Ren, W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data" , IEEE Trans Parallel DistribSyst23(8):1467–1479, (2012)..

[16] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, "Secure ranked keyword search over encrypted cloud data", IEEE International Conference on Distributed Computing Systems(ICDCS), IEEE, Genoa, Italy, (2010), pp. 253-262.

[17] Z. Xia, Y. Zhu, X. Sun and L. Chen, "Secure semantic expansion based search over encrypted cloud data supporting similarity ranking.", Journal of Cloud Computing, Springer 3.1, (2014), pp 1-11.

[18] C. Yang, W. Zhang, J. Xu, N. Yu, "A Fast Privacy-Preserving Multi-keyword Search Scheme on Cloud Data", International Conference on Cloud and Service Computing (CSC). IEEE, Shanghai, China, (2012), pp 104–110.

[19] W. Wong, "Secure KNN Computation on encrypted databases", Proc. SIGMOD, (2009).

[20] S. Zerr , D. Olmedilla, W. Nejdl, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. EDBT '09, 2009.

[21] N. Cao, C. Wang, M. Li, K. Ren, "Privacy-Preserving Multi-Keyword Ranked Search Over Encrypted Cloud Data," IEEE INFOCOM, 2011, pp. 829–37.

[22] C. Wang, N. Cao, J. Li, K. Ren, "Secure Ranked Keyword Search Over Encrypted Cloud Data," Proc. ICDCS '10, 2010.

[23] WK. Wong, DW. Cheung, B. Cao, "Secure KNN Computation on Encrypted Databases," Proc. SIGMOD, 2009.

[24] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM '10 Mini-Conf., San Diego, CA, Mar. 2010.

[25] C. Wang, K. Ren, S. Yu, "Achieving Usable and Privacy-Assured Similarity Search over Outsourced Cloud Data," Proc. IEEE INFOCOM '12, Orlando, FL, Mar. 2012.

[26] M. Li, S Yu, N. Cao, W. Lou, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing," 31st Int'l. Conf. Distributed Computing Systems, 2011, pp. 383–92.

[27] M. Li, S. Yu, K. Ren, Y. Hou, W. Lou, "Toward Privacy-assured and searchable cloud data services", Network, IEEE, 27(4), (2013), pp. 56-62.

[28] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[29] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions,"http://www.techcrunch.com/2006/12/28/gmail-disaster-reports-of-mass-email-deletions/, Dec. 2006.

[30] Amazon.com, "Amazon s3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, 2008.

[31] A. Swaminathan, Y. Mao, G-M. Su, H. Gou, A.L. Vama, S. He, M. Wu and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivality, 2007.

[32] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through PrivacyHomomorphism," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.

[33] S. Gries, "Useful Statics for Corpus Linguistics," A Mosaic of Corpus Linguistics: Selected Approaches, Aquilino Sanchez Moises Almela, eds., pp. 269-291, Peter Lang, 2010.

[34] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," Proc. 41st Ann. ACM Symp. Theory of computing (STOC), pp. 169-178, 2009.

[35] D. Dubin, "The Most Influential Paper Gerard Salton Never Wrote," Library Trends, vol. 52, no. 4, pp. 748-764, 2004.

[36] N. Howgrave-Graham, "Approximate Integer Common Divisors," Proc. Revised Papers from Int'l Conf. cryptography and Lattices (CaLC'01), pp. 51-66, 2001.

[37] M. VanDijk, C. Gentry, S. Halevi, and V. Aikuntanathan, "Fully Homomorphic Encryption over the Integers," Proc. 29th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques, H. Gilbert, pp. 24-43, 2010.

[38] J-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," CRYPTO'11: Proc. 31st Ann. Conf. Advances in Cryptology, 2011.

[39] N. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," Proc. 13th Int'l Conf. Practice and Theory in Public Key Cryptography(PKC), 2010.

[40] P. Golle, J. Staddon, and B. Waters, "Secure Conjuctive Keyword Search over Encrypted Data," Proc. Second Int'l Second Int'l conf. Applied Cryptography and Network Security (ACNS), pp. 31-45, 2004.

[41] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data," Proc. Seventh Int'l Conf. Information and Communications Security (ICICs), 2005.

[42] Cloud Security Alliance, "Top Threats to Cloud Computing," http://www.cloudsecurity alliance.org, 2010.

[43] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li (2013), "Toward secure Multikeyword Top-k retrieval over encrypted cloud data", IEEE transactions on dependable and secure computing, (4), pp. 239-250.