Comparative Study of Different Keyword Searches over Cloud Data

MunavvaraTahaseen[1] and Maniza Hijab[2]

***Abstract* — Cloud computing enables outsourcing of private data onto third party servers where there is a possibility of data breach. For security reasons searchable encryption techniques can be used to encrypt data before outsourcing it to an un-trusted third party cloud server, with the ability to selectively search over it. The objective of ranked search is to improve system usability by searching results based on relevance ranking rather than boolean search. In this paper we briefly provide the different kinds of Ranked searches, the different searchable encryption techniques, which have been proposed so far, compare them and conclude which ranked keyword search is efficient.**

***Keywords* — Cloud, Searchable Encryption, Ranked Search, Trapdoor, Relevance Score.**

I. Introduction

Using cloud computing users can store data on remote servers without the need to have infrastructure. But this raises security issues as the data is hosted on third party servers which may be un-trusted and the data may be sensitive such as e-mails, personal health records, government documents etc. To overcome this problem data owners, encrypt their data before outsourcing onto the cloud. Conventional searching is based on boolean search which is not applicable on cloud as the data is encrypted. Users can search for interested files using keyword search. As data is encrypted, traditional keyword search can't be applied to cloud. Many searchable encryption techniques have been proposed that allow users to securely search over encrypted data through keywords, but they support only boolean search not considering file relevance. Ranked keyword searches, as proposed by [3][4] [5] [8][9] [10] [11] [12] [13] [14] improve searching efficiency by returning files based on their rank as per user relevance. They combine both cryptography and information retrieval techniques.

In this paper, we briefly show the working of different kinds of searches over cloud data, discuss their advantages and disadvantages. Section II provides the background of the different searching techniques proposed so far. In this section we review and compare the different search techniques. In Section III we conclude the paper.

II. Background

Cloud computing enables users to store data on the cloud server. For security issues, the data to be outsourced is encrypted. Different searching techniques have been proposed to securely search over cloud data. The basic idea of all the searches is same. The data owner creates an index for his file collection before encrypting it. After encrypting the file collection, the data owner also encrypts the index file and then outsources the encrypted file collection along with the encrypted index onto the cloud server. The data users search for a file on the cloud server using keywords. The keyword is given to a trapdoor, which encrypts the keyword, which is then searched on the cloud server. The files with the keyword are returned as per their relevance score.

A. Searchable Symmetric Encryption (SSE) [1]:

In [1] index file is created for the file collection and then both the index file and file collection are encrypted. This encrypted file collection and index file are outsourced on to the cloud server. Whenever an end user wants to search the cloud server he does so using keywords. These keywords are encrypted using a trapdoor function and then searched in the cloud server. The search algorithm checks the index file for the existence of the keyword, if the keyword is present, the index file contains the list of file ids which contain the keyword. This list of files will be sent to the end user. The end user decrypts the files to view the contents. But the drawback of this approach is that it can be used only by single user.

B. Non-Adaptive Searchable Symmetric Encryption (SSE-1) [1]:

SSE-1[1] is an improved version of SSE. In [1] the history is generated at once to facilitate the search process. To build index an array and a look up table is used. The user computes both the array and the look up table based on un- encrypted file collection and stores them on the server along with the encrypted file collection [1]. When a user wants to retrieve documents using a keyword, he uses the look up table to find the decryption key and the address for the corresponding entry and sends it to the server [1] [17]. The server then locates and decrypts the given entry and gets the index in the array and the decryption key for the first node of the linked list [1]. As each node of the linked list has information of the next node, the server can decrypt all the nodes [17]. The drawback of this approach is that the process is very complex and the implementation is practically very difficult and is limited to single user search.

C. Adaptive Searchable Symmetric Encryption (SSE -2) [1] :

In SSE-2 [1] the adversary chooses a document collection, receives corresponding index and then receives query word's trapdoor before he chooses the next query word and so on. The index consists of a look up table. For each label in a word w's family an entry is added in look up table, whose value field is identifier of the document that contains an instance of the word w. In order to hide the number of distinct words in each document, the look up table is padded such that the identifier of each document appears in the same number of entries. The drawback of this approach is because of padding the look up table, the search results are not as accurate as SSE-1[1]. The search process is slow as all the labels in a words family have to be searched [1].

D. Multi-User SSE (M-SSE) [1]:

Multi-user SSE uses a broadcast encryption (B.E) scheme with a single user SSE. A group of authorized users receive an encrypted message from B.E centre. The group may change dynamically. Every user has his own secret keys which are used along with a pseudorandom number for decryption. Users when revoked, the data owner takes a new secret key for those users. The additional layer provided by pseudorandom number provides additional security. The drawback of this approach is managing multiple users is an additional overhead.

E. Ranked Searchable Symmetric Encryption (RSSE)[3]:

RSSE follows the framework of searchable symmetric encryption schemes [3]. The data owner preprocesses the data collection to build inverted index. The owner then encrypts the file collection and the index which consists of keyword and relevance score based on keyword frequency. In retrieval phase the user uses an algorithm to generate a trapdoor [2] [3], which takes the keyword from the user and encrypts it. The user submits this encrypted keyword to the cloud server. In [2] [3] the cloud server then returns a list of matched file ID's & their corresponding encrypted relevance scores by searching the index using search Index algorithm. This approach uses Information Retrieval Strategies to compute relevance score. It uses a slightly modified version of Order preserving symmetric encryption OPSE [7] [18] which preserves the numerical ordering of plain text. The use of OPSE over relevance score may enable the adversary to reverse engineer the keyword [2] [3]. To overcome this problem, the authors in [2] [3] have modified OPSE as one-to-many OPSE scheme which uses the unique file ID's together with the plain text as the random score [2] [3]. Due to the use of unique file ID as part of random selection the same plain text will not have the same cipher text. The drawback of RSSE is, it is limited to single keyword search. The range size requires prior-knowledge of the maximum possible duplicates of plain text which is practically very difficult [2] [3].

F. Multi-Keyword Ranked Searchable Encryption (MRSE)[4] [5]:

MRSE [4] [5] uses broadcast encryption [2]. The cloud server searches the index file and returns the results in ranked order. MRSE uses inner product similarity for "co-ordinate matching" [4] [5]. [4] [5] uses a binary data vector to represent the existence of a keyword in the document collection and a binary query vector indicating the existence of keyword in query [4] [5]. The similarity score of document to query is expressed as inner product of query vector and data vector [4] [5]. In [9] the secure k nearest neighbor, Euclidean distance between a database record and a query vector is used to select k nearest database records. MRSE uses inner product of rpi.q, where r is a random number >0 and pi is a database record and q is query vector [4] [5]. MRSE-1 scheme doesn't consider the relationship among similarity scores in different queries [4] [5]. It uses Dimension extending, splitting and encryption procedures for generating the sub index [4] [5]. Splitting and encryption is used to generate trapdoor. Using the trapdoor, cloud server computes the similarity scores of each document [4] [5]. The drawback of this approach is trapdoor privacy is leaked when a user searches two or more times [4] [5]. MRSE 2 preserves trapdoor privacy by breaking the determined relationship between minimal and sub minimal final similarity score and two parameters r and t [4] [5]. In [15] instead of the randomness in the query vector, a dummy keyword is inserted into each data vector and assigned a random value to it. All the vectors are extended to (n+2) dimension instead of (n+1) [15]. The drawback of this approach is due to the inclusion of randomness in similarity score, the final result may not be as accurate MRSE-1 scheme [4] [5].

G. Privacy Preserving Ranked Multi-Keyword search for Multiple Data Owners (PRMSM)[6][16] :

In [6] [16] data owners submit encrypted index to the administration server and the encrypted file collection to the cloud server. The administrator after receiving the encrypted index, re encrypts the index and outsources the same on to the cloud server [6] [16]. Once a data user wants to search for a keyword, he first computes the corresponding trapdoor and submits them to the administration server [6] [16]. The administrative server will further re-encrypt the trapdoors, generates a secret data and submits the re-encrypted trapdoor and the secret key to the cloud server. On receiving the trapdoor, the cloud server searches the encrypted index of each data owner and returns the corresponding set of encrypted files [6] [16]. A data user can also specify the number of relevant files he needs. The administrative server can be any trusted third party like the certificate authority in the public key infra structure [10]. Data user and the administrative server communicate through authentication protocol which consists of five parts to identify the user and ensure that the message is not tampered. Data user prepares his authentication data and encrypts it with a secret key and submits it to the administration server [6] [16]. He generates another secret key and stores both the keys [6] [16]. The administrative server on receiving the encrypted authentication data, decrypts it and if it is authentic generates a new secret key and replies a confirmation data encrypted with the new key, else the administrative sever encrypts the confirmation data with the old key [6] [16]. After getting a reply from the administrative sever the user tries to decrypt it with the second key he generated. If the decrypted data contains the confirmation data, the authentication is successful and the user deletes the second key and considers whether to start another authentication [6] [16]. As multiple data owners are involved in cloud applications [6] [16], they are not interested in sharing the secret keys with others for privacy reasons. Rather they prefer to use their own secret keys to encrypt data [6] [16]. As different data owners use their own secret keys to encrypt their keywords, in [6] [16] the authors propose a scheme where authenticated data users can generate trapdoors without knowing secret keys.

Table I: Comparison of Various Parameters

	SSE-1	SSE-2	M- SSE	RSSE	MRSE-1	MRSE-2
Encryption algorithm	Not specified	Not specified	Broadcast encryption with pseudo-random permutation	One to many OPSE	Broadcast encryption	Broadcast encryption
Access Pattern	Not hidden	Not hidden	Not hidden	Not hidden	Not hidden	Not hidden
Adversaries	Non-adaptive [1]	Adaptive [1]	Non-adaptive [1]	Non-adaptive [1]	Non-adaptive [1]	Non-adaptive [1]
Storage on server	O(n)[1]	O(n)[1]	O(n)[1]	O(n)[1]	O(n)[1]	O(n)[1]
Data structure	Look-up table and an array	Look-up table	Look-up table	Inverted index	Binary data vector	Binary data vector
Pattern matching technique	Equality	Equality	Equality	Equality	Secure KNN using Inner product similarity	Secure KNN using Inner product similarity

n represents the number of documents in the document collection.

The authors have used Information Retrieval techniques to compute relevance scores which are encrypted using an additive order and privacy preserving function [6]. The search results are ranked using the sum of relevance scores. The advantages of this approach is the use of authentication data avoids illegal search and during data user revocation the administration server only needs to update the secret data stored on cloud server [6].Every data owner has his own secret keys for encrypting keywords. Hence one data owner losing his key would not disclose his data [6]. The drawback is that the communication overhead is more as administrative server is involved between the data user and the cloud server and the search process is slow compared to other approaches discussed so far.

III. Conclusion

In this paper, we have analyzed the different types of keyword searches on cloud data proposed so far, and discussed their advantages and disadvantages. In SSE schemes the encryption algorithms to be used have not been discussed and they do not consider the relevance score of results. RSSE introduced the use of ranking in cloud data by the use of relevance score. It is designed only for single keyword search. MRSE schemes work with multiple keywords but for single users. PRMSM is the recent technique proposed to search for multiple keywords from multiple data owners. Compared to MRSE schemes PRMSM takes much less time for index construction, but more time on trapdoor generation as it uses an additional variable to ensure the randomness of trapdoors [6]. Searching is slow in PRMSM due to the additional communication overhead.

References

[1] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions,” Proc. ACM Conf. Computer and Comm. Security (CCS’06), 2006.

[2] C. Wang, N. Cao, J. Li, 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, Aug.2012.

[3] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure Ranked Keyword Search over Encrypted Cloud Data,” Proc. IEEE 30th Int’l Conf. Distributed Computing Systems (ICDCS ’10),2010.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” in Proc.IEEE INFOCOM, Shanghai, China, Apr.2011.

[5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” IEEE Trans. Parallel Distrib. Syst., vol. 25.

[6] Wei Zhang, Y Lin, Sheng Xiao, Jie Wu Siwang Zhou, “Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing” IEEE Transactions on Computers Vol.65, No.5, May 2016.

[7] A. Boldyreva, N. Chenette, and A. O’Neill, “Order- preserving encryption revisited: Improved security analysis and alternative solutions,” in Proc. 31st Annu. Conf. Adv. Cryptol., Aug. 2011.

[8] Cong Wang, Ning Cao, Kui Ren, W Lou “Enabling Secure and Efficient Ranked Keyword Search Over Outsourced CloudData” IEEE Transactions on Parallel and Distributed Systems, vol 23, No.8, August 2012.

[9] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, “Secure knn computation on encrypted databases,” in Proc. of SIGMOD, 2009.

[10] 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.

[11] 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.

[12] Z. Xu, W. Kang, R. Li, K. Yow, and C. Xu, “Efficient multi-keyword ranked query on encrypted data in the cloud,”in Proc. IEEE 19th Int. Conf. Parallel Distrib. Syst., Singapore, Dec. 2012.

[13] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing,” in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010.

[14] Ke Li,, Weiming Zhang, Ce Yang and Nenghai Yu.”Security Analysis on One-to-Many Order Preserving Encryption-Based Cloud Data Search”,IEEE Transactions on Information Forensics and Security,2015.

[15] de.slideshare.net

[16] Wei Zhang, Y Lin, Sheng Xiao, Ting Zhou, Siwang Zhou, "Secure Ranked Multi-keyword Search for Multiple Data Owners in Cloud Computing", 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2014.

[17] www.cs.ucla.edu

[18] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-Preserving Symmetric Encryption," Proc. Int'l Conf. Advances in Cryptology (Eurocrypt '09), 2009.

AUTHORS' PROFILE

Munavvara Tahaseen received Master's Degree M.Tech in Computer Science from Jawaharlal Nehru Technological University, Hyderabad, Telangana, India. Presently she is working as an Assistant Professor in the Department of Information Technology, Muffakham Jah College of Engineering and Technology, Banjara Hills, Hyderabad, India. Her research area of interest includes Cloud Computing, Web Technologies, Data Mining, Big Data.

Maniza Hijab received Master's Degree M.Tech in Computer Science Engineering from Jawaharlal Nehru Technological University. She is presently working as an Associate Professor in the Department of Information Technology, Muffakham Jah College of Engineering and Technology, Banjara Hills, Hyderabad, India. She has more than 20 years of teaching experience. Her area of specialization includes Computer Networking, Database Management System, Cloud Computing, Big Data.