ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Outsourced Cloud Data with Ranked Keyword Search

Shabbir Hussain Shaik[1]
M.Tech Student
Department of Information Technology
V.R Siddhartha Engineering College
Andhra Pradesh – India

Pranathi.K[2]
Assistant Professor
Department of Information Technology
V.R Siddhartha Engineering College
Andhra Pradesh – India

Kranthi.S[3]
Assistant Professor
Department of Information Technology
V.R Siddhartha Engineering College
Andhra Pradesh – India

Abstract: ***In commercial public cloud the cloud computing enables the model of data service outsourcing. In this project, we define and solve the problem of ranked keyword search over cloud data. Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results and further ensures the file retrieval accuracy. We explore the statistical measure approach i.e. relevance score, from information retrieval with focus on ranking and several components of an information retrieval system to build a searchable index.***

Keywords: Ranked search, confidential data, cloud computing.

I. INTRODUCTION

As Cloud Computing becomes widespread, more sensitive information are being centralized into the cloud, such as emails, personal health records, government documents, etc. By storing their data into the cloud, owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. Cloud Computing is the dreamed vision of computing as a utility, where cloud users can store their data into the cloud so as to achieve on-demand high quality applications and services from a shared pool of computing resources [1]. The benefits are not limited: relief of the burden for storage management, information.

Access with independent geographical locations, and avoidance of investment on hardware, software, and personnel maintenances, etc [2]. As Cloud Computing becomes prevalent, more sensitive information are being under control into the cloud, such as company finance information, government documents, personal health records, and emails etc. The fact that cloud server and information owners are no longer in the same trusted domain may put the outsourced unencrypted data at risk [3]: the cloud server may even be hacked [5] or may dispose data information to unauthorized entities [4]. It follows that sensitive data has to be encrypted before outsourcing for data privacy and combating unsolicited accesses. One of the most familiar ways to do so is through keyword search. Data encryption makes efficient data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud, data owners share their data with users, who might want to only retrieve specific data files they are interested in during a given particular session. Such keyword search technique allows users to selectively retrieve files of interest and has been enforced in plaintext search [6]. Unfortunately, data encryption, restricts user's ability to perform keyword search and demands the protection for keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data. Although traditional searchable encryption schemes (e.g. [7]–[11], to list a few) allow a user to securely search over encrypted data through keywords without decrypting it, these techniques support conventional Boolean keyword search, without capturing any relevance of the files in the search result. When enforced in large collaborative data outsourcing cloud platform, they may suffer from the following two main drawbacks. On the other

hand, invariably sending back all files solely based on presence/absence of the keyword incurs unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud paradigm. For each search request, users without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to match ones interest which demands large amount of post processing over-head; lacking of effective mechanisms to ensure the file retrieval accuracy is a significant drawback of existing searchable encryption schemes in Cloud Computing. The state of the art in information retrieval (IR) community has already been utilizing various scoring mechanisms [13] to quantify and rank-order the relevance of files in response to any given search query. the importance of ranked search has received attention for a long history in the context of plaintext searching by Information Retrieval (IR) community, it is still being overlooked and is addressed in encrypted data search. Therefore, how to alter a searchable encryption system with support of secure ranked search, is the problem in this paper. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer towards deployment of privacy-preserving data hosting services in the context of Cloud Computing. To achieve our design goals on both system security and usability, we propose to bring together the advance of both crypto and IR community to design the ranked searchable symmetric encryption scheme, in the spiritof "as-strong as-possible" security guarantee. we explore the statistical measure approach from IR and text-mining to embed weight information (i.e. relevance score) of each file during the establishment of searchable index before outsourcing the encrypted file collection. As directly outsourcing relevance scores will dispose lots of sensitive frequency information against the keyword privacy, we then combine a recent crypto primitive [14] order preserving symmetric encryption (OPSE) and properly modify it to develop a one-to-many order-preserving mapping technique for our purpose to protect those sensitive weight information, while providing efficient ranked search functionalities. Our contribution can be summarized as follows:

1) For the first time, we define the problem of secure ranked keyword search over encrypted cloud data, and provide such an effective protocol, which fulfills the secure ranked search functionality with little relevance score information leakage against keyword privacy.

2) Thorough security analysis shows that our ranked searchable symmetric encryption scheme indeed enjoys "as-strong-as-possible" security guarantee compared to previous SSE schemes.

3) We investigate the practical considerations and enhancements of our ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of our proposed one-to many order-preserving mapping technique.

4) Even though Relevance scores are neglected, Word count percentage calculation is done by calculating no of words to total no of words.

5) We have generated dynamically created Trapdoor Random function, i.e Random key generation with variable length for 'n' no of users.

6) We have used cloud Microsoft Onedrive as our Cloud Environment, which is a real time Environment.

7) We have developed User Ranking that is user can have ability to change Rank of file.

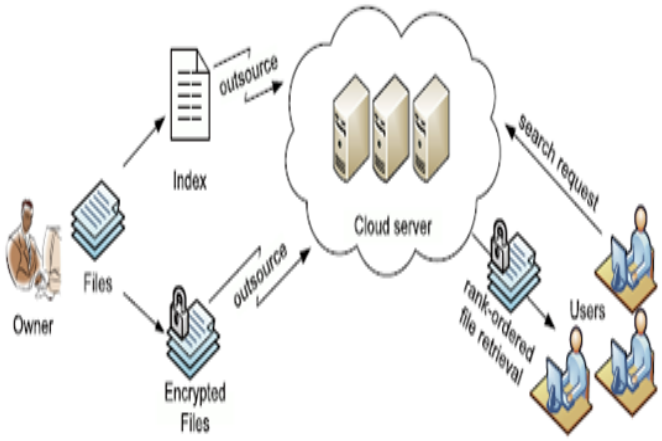

Fig 1: Architecture for search over encrypted cloud data

II. PROBLEM STATEMENT

Traditional searchable encryption schemes allow a user to securely search over encrypted data through keywords without first decrypting it, these techniques support only conventional Boolean keyword search without capturing any relevance of the files in the search result. For each search request, users without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to find ones most matching their interest, which demands possibly large amount of post processing overhead; when directly enforced in large collaborative data outsourcing cloud environment, they may suffer from the following two main drawbacks. Invariably sending back all files solely based on presence/absence of the keyword further incurs large unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud environment.

Our work is among the first few ones to explore ranked search over encrypted data in Cloud Computing. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer toward practical deployment of privacy-preserving data hosting services in the context of Cloud Computing. To achieve our design goals on both system security and usability, we propose to bring together the advance of both crypto and Information Retrieval (IR) community to design the ranked searchable symmetric encryption (RSSE) scheme, in the spirit of "as-strong-as-possible" security guarantee.

A. The System and Threat Model

We consider an encrypted cloud data hosting service involving three different entities, as illustrated in Fig. 1: data owner, data user, and cloud server. Data owner has a collection of n data files C = (F1, F2,…,Fn) that he wants to outsource on the cloud server in encrypted form while still keeping the capability to search through them for effective data utilization reasons. To do so, before outsourcing, data owner have to build a secure searchable index I from a set of m distinct keywords W = (w1,w2,….,wm) extracted from the file collection C, and store both the index I and the encrypted file collection C on the cloud server.

We assume the authorization between the data owner and users is appropriately done. To search the file collection for a given keyword w, an authorized user generates and submits a search request in a secret form a trapdoor Tw of the keyword w to the cloud server. Upon receiving the search request T_w, the cloud server is responsible to search the index I and return the corresponding set of files to the user. We consider the ranked keyword in secure search, problem as follows: the search result should be returned according to certain ranked relevance criteria (e.g., keyword frequency based scores, as will be introduced shortly), to improve file retrieval accuracy for users without prior knowledge on the file collection C. However, cloud server should learn nothing or little about the relevance criteria as they exhibit significant sensitive information against keyword

privacy. To reduce bandwidth, the user may send an optional value k along with the trapdoor T_w and cloud server only sends back the top-k most relevant files to the user's interested keyword w.

B. Design Goals

To alter ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model, our system design should achieve the following security and performance guarantee. Specifically, we have the following goals:

- Ranked keyword search: to explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework;
- Security guarantee: to achieve the " strong- as-possible" security strength compared to existing searchable encryption schemes prevent cloud server from learning the plaintext of either the data files or the searched keywords, and;
- Efficiency: above goals should be achieved with minimum communication and computation overhead.

III. SECURE RANKED SEARCHABLE SYMMETRIC ENCRYPTION SCHEME

The above approach demonstrates the core problem that causes the inefficiency of ranked searchable encryption. That is how to let server quickly perform the ranking without actually knowing the relevance scores. To effectively support ranked search over encrypted file collection, we now resort to the newly developed cryptographic primitive – order preserving symmetric encryption (OPSE) [14] to achieve more practical performance. as we now let server know the relevance order. However, this is the information we want to tradeoff for efficient RSSE. Note that by resorting to OPSE, our security guarantee of RSSE is inherently weakened compared to SSE, We will first briefly discuss the primitive of OPSE and its pros and cons. Then we show how we can adapt it to suit our purpose for ranked searchable encryption with an "as-strong-as-possible" security guarantee. Finally, we demonstrate how to choose different scheme parameters via concrete examples.

IV. SECURITY ANALYSIS

We solve the security of the proposed scheme by analyzing its fulfilment of the security guarantee described in previous. the cloud server should not learn the plaintext of either the data files or the searched keywords. We start from the security analysis of our one-to-many order-preserving mapping. Then we analyze the security strength of the combination of one to many order-preserving mapping and SSE.

A. Security Analysis for One-to-many Mapping

Our one-to-many order-preserving mapping is adapted from the original OPSE, by introducing the file ID as the additional seed in the final cipher text chosen process. Since such adaptation only functions at the final cipher text selection process, it has nothing to do with the randomized plaintext-to-bucket mapping process in the original OPSE.

B. Security Analysis for Ranked Keyword Search

Compared to the original SSE, the new scheme embeds the encrypted relevance scores in the searchable index in addition to file ID. Due to the security strength of the file encryption scheme, the file content is clearly well protected. Thus, we only need to focus on keyword privacy. Thus the encrypted scores are the only additional information that the opponent can utilize against the security guarantee, i.e., keyword privacy and file confidentiality. From previous discussion, we know that as long as data owner properly chooses the range size R sufficiently large, the encrypted scores in the searchable index will only be a sequence of order-preserved numeric values with very low duplicates.

V. Performance Analysis

We conducted a thorough experimental evaluation of the proposed techniques on real data set: Request for comments database (RFC) [23]. At the time of writing, the RFC database contains 6583 plain text entries and totals about 459 MB. This file set contains a large number of technical keywords, many of which are unique to the files in which they are discussed.

The performance of our scheme is evaluated regarding the effectiveness and efficiency of our proposed one-to many order-preserving mapping, as well as the overall performance of our RSSE scheme, including the cost of index construction as well as the time necessary for searches. Our experiment is conducted using Java programming language on a Windows machine with Intel i3 CPU running at 2.20GHz. Algorithms use both open ssl and MATLAB libraries. Note that though we use a single server in the experiment, in practice we can separately store the searchable index and the file collections on different virtualized service nodes in the commercial public cloud, such as the Amazon EC2 and Amazon S3, respectively. In that way, even if data owners choose to store their file collection in different geographic locations for increased availability, the underlying search mechanism, which always takes place based on the searchable index, will not be affected at all.

VI. Related Work

Searchable Encryption: Traditional searchable encryption [8]–[12], has been widely studied as a cryptographic primitive, with a focus on security definition formalizations and efficiency improvements. Songet al. [8] first introduced the notion of searchable encryption. They proposed a scheme in the symmetric key setting, where each word in the file is encrypted independently under a special two-layered encryption construction. Thus, a searching overhead is linear to the whole file collection length. Goh [9] developed a Bloom filter based per-file index, reducing the work load for each search request proportional to the number of files in the collection. Chang et al. [11] also developed a similar per-file index scheme. To further enhance search efficiency, Curtmola et al. [12] proposed a per-keyword based approach, where a single encrypted hash table index is built for the entire file collection, with each entry consisting of the trapdoor of a keyword and an encrypted set of related file identifiers. Searchable encryption has also been considered in the public-key setting. Boneh et al. [10] presented the first public-key based searchable encryption scheme, with an analogous scenario to that of [8]. In their construction, anyone with the public key can write to the data stored on the server but only authorized users with the private key can search. Recently, aiming at tolerance of both minor typos and format in consistencies in the user search input, fuzzy keyword search over encrypted cloud data Note that all these schemes support only Boolean keyword search, and none of them support the ranked search problem which we are focusing in this paper propose a privacy-preserving multi-keyword ranked search scheme, which extends our previous work in [1] with support of multi-keyword query. They choose the principle of “coordinate matching”, i.e., as many matches as possible, to capture the similarity between a multi keyword search query and data documents, and later quantitatively formalize the principle by a secure inner] product computation mechanism. One disadvantage of the scheme is that cloud server has to linearly traverse the whole index of all the documents for each search request, while ours is as efficient as existing SSE schemes with only constant search cost on cloud server .Secure top-k retrieval from Database Community [18] from database community are the most related work to our proposed RSSE. The idea of uniformly distributing posting elements using an order-preserving cryptographic function. However, the order-preserving mapping function proposed does not support score dynamics, i.e., any insertion and updates of the scores in the index will result in the posting list completely rebuilt. Besides, when scores following different distributions need to be inserted, their score transformation function still needs to be rebuilt. On the contrary, in our scheme the score dynamics can be gracefully handled, which is an important benefit inherited from the original OPSE. It uses a different order-preserving mapping based on pre-sampling and training of the relevance scores to be outsourced, which is not as efficient as our proposed schemes. This can be observed from the Linear Search procedure in Algorithm 1, where the same score will always be mapped to the same random-sized non-overlapping bucket, given the same encryption key. In other words, the newly changed scores will not affect previous mapped values. We note that supporting score dynamics, which can save quite a lot of computation overhead when file

collection changes, is a significant advantage in our scheme. Moreover, both works above do not exhibit thorough security analysis which we do in the paper. Other Related Techniques Allowing range queries over encrypted data in the public key settings where advanced privacy preserving schemes were proposed to allow more sophisticated multi-attribute search over encrypted files while preserving the attributes' privacy. Moreover, the two schemes do not support the ordered result listing on the server side. Thus, they cannot be effectively utilized in our scheme since the user still does not know which retrieved files would be the most relevant. Though these two schemes provide provably strong security, they are generally not efficient in our settings, as for a single search request, a full scan and expensive computation over the whole encrypted scores corresponding to the keyword posting list are required.

VII. CONCLUSION

Ranked keyword search on remotely stored data is done by saving files in cloud and retrieve the files by searching through the keywords. Retrieved files are presented in ranked order which is done by using ranking algorithm in the index page. Security for data stored in cloud is done trough saving encrypted files and privacy of data is maintained by providing different trapdoors to different users. Ranked analysis is done by score dynamics i.e taking the user choices into consideration and giving highest rank to user chosen file so that user can get more efficient results. Investigate some further enhancements of our ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of our proposed one-to-many order-preserving mapping technique. Through thorough security analysis, we show that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of our solution.

References

1. P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on Jan. 23rd, 2010 Online at http://csrc. nist.gov/groups/SNS/cloud-computing/index.html, 2010.
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCBEECS- 2009-28, Feb 2009.
3. Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www. cloudsecurityalliance.org.
4. Z. Slocum, "Your google docs: Soon in search results?" http:// news.cnet.com/8301-17939 109-10357137-2.html, 2009.
5. B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at http://voices.washingtonpost.com/securityfix/2009/01/ payment processor breach may b.html, Jan. 2009.
6. I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.
7. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of IEEE Symposium on Security and Privacy'00, 2000.
8. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/.
9. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYP'04, volume 3027 of LNCS. Springer, 2004.
10. Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS'05, 2005.
11. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS'06, 2006.
12. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS'06, 2006.
13. A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.
14. A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Orderpreserving symmetric encryption," in Proc. of Eurocrypt'09, volume 5479 of LNCS. Springer, 2009.
15. J. Zobel and A. Moffat, "Exploring the similarity space," SIGIR Forum, vol. 32, no. 1, pp. 18–34, 1998.
16. O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM, vol. 43, no. 3, pp. 431–473, 1996.
17. M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Proc. of Crypto'07, volume 4622 of LNCS. Springer, 2007.
18. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k retrieval from a confidential index," in Proc. of EDBT'09, 2009.

