IPHV8I2005X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EFFICENT KEYWORD SEARCH AND SHARING ENCRYPTED DATA IN CLOUD SERVICES

[#1]GADDAM SANKEERTHANA- M.Tech Student,
[#2]K.CHANDRASENA CHARY-Associate Professor,
Department of CSE
SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES,KARIMNAGAR,T.S, INDIA.

ABSTRACT—As Cloud Computing becomes prevalent, sensitive information are being increasingly centralized into the cloud. For the protection of data privacy, sensitive data has to be encrypted before outsourcing, which makes effective data utilization a very challenging task. Although traditional searchable encryption schemes allow users to securely search over encrypted data through keywords, these techniques support only boolean search, without capturing any relevance of data files. This approach suffers from two main drawbacks when directly applied in the context of Cloud Computing. On the one hand, users, who do not necessarily have pre-knowledge of the encrypted cloud data, have to post process every retrieved file in order to find ones most matching their interest; On the other hand, invariably retrieving all files containing the queried keyword further incurs unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud paradigm. In this paper, for the first time we define and solve the problem of effective yet secure ranked keyword search over encrypted cloud data. 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 practical deployment of privacy-preserving data hosting services in Cloud Computing. We first give a straightforward yet ideal construction of ranked keyword search under the state-of-the-art searchable symmetric encryption (SSE) security definition, and demonstrate its inefficiency. To achieve more practical performance, we then propose a definition for ranked searchable symmetric encryption, and give an efficient design by properly utilizing the existing cryptographic primitive, order-preserving symmetric encryption (OPSE). Thorough analysis shows that our proposed solution enjoys "as-strong-as possible" security guarantee compared to previous SSE schemes, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of the proposed solution.

I.INTRODUCTION

Cloud Computing enables cloud customers to remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [1]. The benefits brought by this new computing model include but are not limited to: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2]. With the prevalence of cloud services, more and more sensitive information are being centralized into the cloud servers, such as emails, personal health records, private videos and photos, company finance data, government documents, etc [3]. To protect data privacy and combat unsolicited accesses, sensitive data has to be encrypted before outsourcing [4] so as to provide end-to-end data confidentiality assurance in the cloud and beyond. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios [5]. Unfortunately, data encryption, which restricts user's ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data. Although traditional searchable encryption schemes (e.g. [6]–[10], to list a few) allow a user to securely search over encrypted data through keywords without first decrypting it, these techniques support only conventional Boolean keyword search1, without capturing any relevance of the files in the search result. When directly applied in large collaborative data outsourcing cloud environment, they may suffer from the following two main drawbacks. On the one hand, 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 postprocessing overhead; On the other hand, 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 paradigm. In short, lacking of effective mechanisms to ensure the file retrieval accuracy is a significant drawback of existing searchable encryption schemes in the context of Cloud Computing. Nonetheless,

the state-of-the-art in information retrieval (IR) community has already been utilizing various scoring mechanisms [11] to quantify and rank-order the relevance of files in response to any given search query. Although the importance of ranked search has received attention for a long history in the context of plaintext searching by IR community, surprisingly, it is still being overlooked and remains to be addressed in the context of encrypted data search. Therefore, how to enable a searchable encryption system with support of secure ranked search, is the problem tackled in this paper. 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 towards 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 IR community to design the ranked searchable symmetric encryption scheme, in the spirit of "as-strong-as-possible" security guarantee. Specifically, 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 leak lots of sensitive frequency information against the keyword privacy, we then integrate a recent crypto primitive [12] order preserving symmetric encryption (OPSE) and properly modify it 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-strongas-possible" security guarantee compared to previous SSE schemes.

3) Extensive experimental results demonstrate the effectiveness and efficiency of the proposed solution.

The rest of the paper is organized as follows. Section II gives the system and threat model, our design goals, notations and preliminaries. Then we provide the framework, definitions and basic scheme in Section III, followed by Section IV, which gives the detailed description of our ranked searchable symmetric encryption system. Section V and VI gives the security analysis and performance evaluation, respectively. Related work for both searchable encryption and secure result ranking is discussed in Section VII. Finally, Section VIII gives the concluding remark of the whole paper.

II. RELATED WORK

Searchable Encryption: Traditional searchable encryption [6]–[10] has been widely studied as a cryptographic primitive, with a focus on security definition formalizations and efficiency improvements. Song et al. [6] 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 [7] 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. [9] also developed a similar per-file index scheme. To further enhance search efficiency, Curtmola et al. [10] 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. [8] presented the first public-key based searchable encryption scheme, with an analogous scenario to that of [6]. 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. As an attempt to enrich query predicates, conjunctive keyword search over encrypted data have also been proposed in [19]–[21]. Very recently, aiming at tolerance of both minor typos and format inconsistencies in the user search input, fuzzy keyword search over encrypted cloud data has been proposed by Li. et al in [22]. 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. Secure top-k retrieval from Database Community [16], [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 was first discussed in [18]. However, the order-preserving mapping function proposed in [18] 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. [16] 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. 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. This can be observed from the Binary Search (·) procedure in Algorithm 1, where the same score will always be mapped to the same random-sized no 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.

III. PROBLEM STATEMENT

A. The System and Threat Model

We consider a cloud data hosting service involving three different entities, as illustrated in Fig. 1: data owner (O), Owner outsource Files outsource Encrypted Files search request rank file retrieval -ordered Index Users Cloud server Fig. 1: Architecture of the search over encrypted cloud data data user (U), and cloud server (CS). 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 will first build a secure searchable index I from a set of m distinct keywords W = (w1, w2, ..., wm) extracted2 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 Tw, the cloud server is responsible to search the index I and return the corresponding set of files to the user. We consider the secure ranked keyword 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 themselves as they exhibit significant sensitive information against keyword privacy. To reduce bandwidth, the user may send an optional value k along with the trapdoor Tw and cloud server only sends back the top-k most relevant files to the user's interested keyword w. We consider an "honest-but-curious" server in our model, which is consistent with most of the previous searchable encryption schemes. We assume the cloud server acts in an "honest" fashion and correctly follows the designated protocol specification, but is "curious" to infer and analyze the message flow received during the protocol so as to learn additional information. In other words, the cloud server has no intention to actively modify the message flow or disrupt any other kind of services.

B. Design Goals

To enable ranked searchable symmetric encryption for effective utilization of outsourced cloud data under the aforementioned model, our system design should achieve the following security and performance guarantee. Specifically, we have the following goals: i) Ranked keyword search: to explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework; ii) Security guarantee: to prevent cloud server from learning the plaintext of either the data files or the searched keywords, and achieve the as-strong-as-possible security strength compared to the existing searchable encryption schemes; iii) Efficiency: above goals should be achieved with minimum communication and computation overhead.

C. Notation and Preliminaries

• C – the file collection to be outsourced, denoted as a set of n data files C = (F1, F2,...,Fn).
• W – the distinct keywords extracted from file collection C, denoted as a set of m words W = (w1, w2, ..., wm).
• id(Fj) – the identifier of file Fj that can help uniquely locate the actual file.
• I – the index built from the file collection, including a set of posting lists {I(wi)}, as introduced below.
• Twi – the trapdoor generated by a user as a search request of keyword wi.
• F(wi) – the set of identifiers of files in C that contain keyword wi.
• Ni – the number of files containing the keyword wi and Ni = |F(wi)|.

We now introduce some necessary information retrieval background for our proposed scheme:

Inverted Index In information retrieval, inverted index (also referred to as postings file) is a widely used indexing structure that stores a list of mappings from keywords to the corresponding set of files that contain this keyword, allowing full text search [11]. For ranked search purposes, the task of determining which files are most relevant is typically done by assigning a numerical score, which can be precomputed, to each file based on some ranking function introduced below. One example posting list of an index is shown in Fig. 2. We will use this inverted index structure to give our basic ranked searchable symmetric encryption construction. Ranking Function In information retrieval, a ranking function is used to calculate relevance scores of matching files to a given search request. The most widely used statistical measurement for evaluating relevance score in the information retrieval community uses the TF × IDF rule, where TF (term frequency) is simply the number of times a given term or keyword (we will use them interchangeably hereafter) appears within a file (to measure the importance of the term within the particular file), and IDF (inverse document frequency) is obtained by dividing

the number of files in the whole collection by the number of files containing the term (to measure the overall importance of the term within the whole collection). Among several hundred variations of the TF × IDF weighting scheme, no single combination of them outperforms any of the others universally [13]. Thus, without loss of generality, we choose an example formula that is commonly used and widely seen in the literature (see Chapter 4 in [5]) for the relevance score calculation in the following presentation.

Word	wi				
File ID	Fi1	Fi2	Fi3		FiN
Relevance Score	6.52	2.29	13.42	4.76	13.80

Its definition is as follows:

$$Score(Q, Fd) = t \in Q \; 1 |Fd| \cdot (1 + \ln fd,t) \cdot \ln(1 + N \; ft \;). \quad (1)$$

Here Q denotes the searched keywords; fd,t denotes the TF of term t in file Fd; ft denotes the number of files that contain term t; N denotes the total number of files in the collection; and |Fd| is the length of file Fd, obtained by counting the number of indexed terms, functioning as the normalization factor.

IV. THE DEFINITIONS AND BASIC SCHEME

In the introduction we motivated the ranked keyword search over encrypted data to achieve economies of scale for Cloud Computing. In this section, we start from the review of existing searchable symmetric encryption (SSE) schemes and provide the definitions and framework for our proposed ranked searchable symmetric encryption (RSSE). Note that by following the same security guarantee of existing SSE, it would be very inefficient to support ranked search functionality over encrypted data, as demonstrated in our basic scheme. The discussion of its demerits will lead to our proposed scheme.

A. Background on Searchable Symmetric Encryption

Searchable encryption allows data owner to outsource his data in an encrypted manner while maintaining the selectivelysearch capability over the encrypted data. Generally, searchable encryption can be achieved in its full functionality using an oblivious RAMs [14]. Although hiding everything during the search from a malicious server (including access pattern), utilizing oblivious RAM usually brings the cost of logarithmic number of interactions between the user and the server for each search request. Thus, in order to achieve more efficient solutions, almost all the existing work on searchable encryption literature resort to the weakened security guarantee, i.e., revealing the access pattern and search pattern but nothing else. Here access pattern refers to the outcome of the search result, i.e., which files have been retrieved. The search pattern includes the equality pattern among the two search requests (whether two searches were performed for the same keyword), and any information derived thereafter from this statement. We refer readers to [10] for the thorough discussion on SSE definitions.

Having a correct intuition on the security guarantee of existing SSE literature is very important for us to define our ranked searchable symmetric encryption problem. As later we will show that following the exactly same security guarantee of existing SSE scheme, it would be very inefficient to achieve ranked keyword search, which motivates us to further weaken the security guarantee of existing SSE appropriately (leak the relative relevance order but not the relevance score) and realize an "as-strong-as-possible" ranked searchable symmetric encryption. Actually, this notion has been employed by cryptographers in many recent work [12], [15] where efficiency is preferred over security.

B. Definitions and Framework of RSSE System

We follow the similar framework of previously proposed searchable symmetric encryption scheme [10] and adapt the framework for our ranked searchable encryption system. A ranked searchable encryption scheme consists of four algorithms (KeyGen, Build Index, TrapdoorGen, and Search Index). And our ranked searchable encryption system can be constructed from these four algorithms in two phases—Setup and Retrieval:

Setup: The data owner initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file collection C by using BuildIndex to generate the searchable index from the unique words extracted from C. The owner then encrypts the data file collection C, and publishes the index including the keyword frequency based relevance scores in some encrypted form, together with the encrypted collection C to the Cloud. As part of Setup phase, the data owner also needs to distribute the necessary secret parameters (in our case, the trapdoor generation key) to a group of authorized users by employing off-the-shelf public key cryptography or more efficient primitive such as broadcast encryption.

Retrieval: The user uses TrapdoorGen to generate a secure trapdoor corresponding to his interested keyword, and submits it to the cloud server. Upon receiving the trapdoor, the cloud server will derive a list of matched file IDs and their corresponding encrypted relevance scores by searching the index via SearchIndex. The matched files should be sent back in a ranked sequence based on the relevance scores. However, the server should learn nothing or little beyond the order of the relevance scores. Note that in our design, we focus on single keyword search. In this case, the IDF factor in equation 1 is always constant with regard to the given searched keyword. Thus, search results can be accurately ranked based only on the term frequency and file length information contained within the single file using equation 2:

Score(t, Fd) = 1 |Fd| · (1 + ln fd,t). (2)

Data owner can keep a record of these two values and precalculate the relevance score, which introduces little overhead regarding to the index building. We will demonstrate this via experiments in the performance evaluation Section VI.

V. CONCLUSION

In this paper, as an initial attempt, we motivate and solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. We first give a basic scheme and show that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked search. We then appropriately weaken the security guarantee, resort to the newly developed crypto primitive OPSE, and derive an efficient one-to-many order-preserving mapping function, which allows the effective RSSE to be designed. 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. Following the current research, we propose several possible directions for future work on ranked keyword search over encrypted data. The most promising one is the support for multiple keywords. In this case, for the security requirement of searchable encryption, constructions for conjunctive keyword search in the existing literature [19]–[21] might be good candidates for our proposed ranked search. However, as the IDF factor now has to be included for score calculation, new approaches still need to be designed to completely preserve the order when summing up scores for all the provided keywords. Another interesting direction is to integrate advanced crypto techniques, such as attribute-based encryption to enable finegrained access control in our multi-user settings.

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. UCB-EECS-2009-28, Feb 2009.

[3] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010, January 2010.

[4] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

[5] I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.

[6] 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.

[7] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/.

[8] 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.

[9] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS'05, 2005.

[10] 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.

[11] A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.

[12] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in Proceedings of Eurocrypt'09, volume 5479 of LNCS. Springer, 2009.

[13] J. Zobel and A. Moffat, "Exploring the similarity space," SIGIR Forum, vol. 32, no. 1, pp. 18–34, 1998.

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

[15] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Proceedings of Crypto'07, volume 4622 of LNCS. Springer, 2007.

[16] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k retrieval from a confidential index," in Proceedings of EDBT'09, 2009.

[17] RFC, "Request For Comments Database," http://www.ietf.org/rfc.html.

[18] 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 Proc. of the Workshop on Storage Security and Survivability, 2007.

[19] P. Golle, J. Staddon, and B. R. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," in Proc. of ACNS'04, 2004, pp. 31–45.

[20] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in Proc. of ICICS'05, 2005.