Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data

Ananya Veeresh, Meghana D , Sulekha M S
Dept. of Computer Science & Engineering
GSSSIETW, Mysuru

Abstract—With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, for the first time, we define and solve the challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE).We establish a set of strict privacy requirements for such a secure cloud data utilization system. Among various multi keyword semantics, we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query.

I. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where cloud customers can 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. With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data has to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. The trivial solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of bandwidth cost in cloud scale systems. Moreover, aside from eliminating the local storage management, storing data into the cloud serves no purpose unless they can be easily searched and utilized. Thus, exploring privacy-preserving and effective search service over encrypted cloud data is of paramount importance. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability.

.

II. PROBLEM FORMULATION

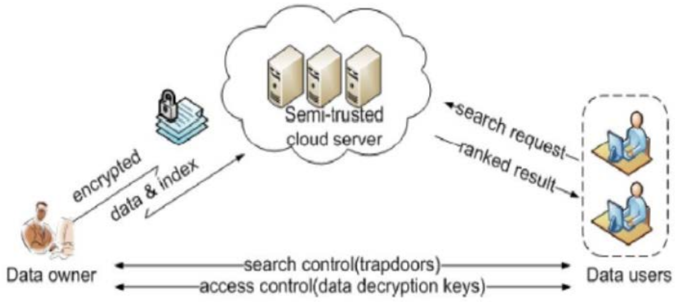

Fig 1: Architecture of the search over encrypted cloud data

A. System Model

Considering a cloud data hosting service involving three different entities, as illustrated in Fig. 1: the data owner, the data user, and the cloud server. The data owner has a collection of data documents *F* to be outsourced to the cloud server in the encrypted form *C*. To enable the searching capability over *C* for effective data utilization, the data owner, before outsourcing, will first build an encrypted searchable index *I* from *F*, and then outsource both the index *I* and the encrypted document collection *C* to the cloud server. To search the document collection for *t* given keywords, an authorized user acquires a corresponding trapdoor *T* through search control mechanisms, e.g., broadcast encryption [8]. Upon receiving *T* from a data user, the cloud server is responsible to search the index *I* and return the corresponding set of encrypted documents. To improve the document retrieval accuracy, the search result should be ranked by the cloud server according to some ranking criteria (e.g., coordinate matching, as will be introduced shortly). Moreover, to reduce the communication cost, the data user may send an optional number *k* along with the trapdoor *T* so that the cloud server only sends back top-*k* documents that are most relevant to the search query. Finally, the access control mechanism [24] is employed to manage decryption capabilities given to users.

B. Threat Model

The cloud server is considered as "honest-but-curious" in our model, which is consistent with related works on cloud security [24], [25]. Specifically, the cloud server acts in an "honest" fashion and correctly follows the designated protocol specification. However, it is "curious" to infer and analyze data (including index) in its storage and message flows received during the protocol so as to learn additional information. Based on what information the cloud server knows, we consider two threat models with different attack capabilities as follows.

Known Cipher text Model In this model, the cloud server is supposed to only know encrypted dataset *C* and searchable index *I*, both of which are outsourced from the data owner.

Known Background Model In this stronger model, the cloud server is supposed to possess more knowledge than what can be accessed in the known ciphertext model. Such information may include the correlation relationship of given search requests (trapdoors), as well as the dataset related statistical information.

C. Notations

- *F* – the plaintext document collection, denoted as a set of *m* data documents $F = (F1, F2, \ldots, Fm)$.
- *C* – the encrypted document collection stored in the cloud server, denoted as $C = (C1, C2, \ldots, Cm)$.
- *W* – the dictionary, i.e., the keyword set consisting of *n* keyword, denoted as $W = (W1, W2, \ldots, Wn)$.
- *I* – the searchable index associated with *C*, denoted as $(I1, I2, \ldots, Im)$ where each subindex *Ii* is built for *Fi*.
- f*W* – the subset of *W*, representing the keywords in a search request, denoted as $\text{f}W = (Wj1, Wj2, \ldots, Wjt)$.
- *T*f*W* – the trapdoor for the search request f*W*.
- *F*f*W* – the ranked id list of all documents according to their relevance to f*W*.

III. FRAMEWORK AND PRIVACY REQUIREMENTS FOR MRSE

In this section, we define the framework of multi-keyword ranked search over encrypted cloud data (MRSE).

A. MRSE Framework

For easy presentation, operations on the data documents are not shown in the framework since the data owner could easily employ the traditional symmetric key cryptography to encrypt and then outsource data. With focus on the index and query,the MRSE system consists of four algorithms as follows.

- Setup(1ℓ) Taking a security parameter ℓ as input, the data owner outputs a symmetric key as SK.
- BuildIndex(F, SK) Based on the dataset F, the data owner builds a searchable index I which is encrypted by the symmetric key SK and then outsourced to the cloud server. After the index construction, the document collection can be independently encrypted and outsourced.
- Trapdoor(fW) With t keywords of interest in fW as input,this algorithm generates a corresponding trapdoor TfW.
- Query(TfW, k, I) When the cloud server receives a query request as (TfW, k), it performs the ranked search on the index I with the help of trapdoor TfW, and finally returns FfW, the ranked id list of top-k documents sorted by their similarity with fW.

Neither the search control nor the access control is within the scope of this paper. While the former is to regulate how authorized users acquire trapdoors, the later is to manage users' access to outsourced documents.

B. Privacy Requirements for MRSE

As for the data privacy, the data owner can resort to the traditional symmetric key cryptography to encrypt the data before outsourcing, and successfully prevent the cloud server from prying into the outsourced data. With respect to the index privacy, if the cloud server deduces any association between keywords and encrypted documents from index, it may learn the major subject of a document, even the content of a short document [26]. Therefore, the searchable index should be constructed to prevent the cloud server from performing such kind of association attack.

Keyword Privacy As users usually prefer to keep their search from being exposed to others like the cloud server, the most important concern is to hide what they are searching, i.e., the keywords indicated by the corresponding trapdoor. Although the trapdoor can be generated in a cryptographic way to protect the query keywords, the cloud server could do some statistical analysis over the search result to make an estimate. As a kind of statistical information, document frequency (i.e., the number of documents containing the keyword) is sufficient to identify the keyword with high probability [27]. When the cloud server knows some background information of the dataset, this keyword specific information may be utilized to reverse-engineer the keyword.

Trapdoor Unlinkability The trapdoor generation function should be a randomized one instead of being deterministic. In particular, the cloud server should not be able to deduce the relationship of any given trapdoors, e.g., to determine whether the two trapdoors are formed by the same search request. Otherwise, the deterministic trapdoor generation would give the cloud server advantage to accumulate frequencies of different search requests regarding different keyword(s), which may further violate the aforementioned keyword privacy requirement. So the fundamental protection for trapdoor is to introduce sufficient nondeterminacy into the trapdoor generation procedure.

IV. PERFORMANCE ANALYSIS

A. Precision and Privacy

As some dummy keywords are inserted into each data vector and some of them are selected in every query. Therefore, similarity scores of documents will be not exactly accurate. In other words, when the cloud server returns top-k documents based on similarity scores of data vectors to query vector, some of real top-k relevant documents for the query may be excluded. This is because either their original similarity scores are decreased or the similarity scores of some documents out of the real top-k are increased, both of which are due to the impact of dummy keywords inserted into data vectors.

B. Efficiency

To build a searchable subindex Ii for each document Fi in the dataset F, the first step is to map the keyword set extracted from the document Fi to a data vector Di, followed by encrypting every data vector. The time cost of mapping or encrypting depends directly on the dimensionality of data vector which is determined by the size of the dictionary, i.e., the number of indexed keywords. And the time cost of building the whole index is also related to the number of subindex which is equal to the number of documents in the dataset.

V. RELATED WORK

A. Single Keyword Searchable Encryption

Traditional single keyword searchable encryption schemes [5]–[13], [22] usually build an encrypted searchable index such that its content is hidden to the server unless it is given appropriate trapdoors generated via secret key(s) [2]. It is first studied by Song et al. [5] in the symmetric key setting, and improvements and advanced security definitions are given in Goh [6], Chang et al. [7] and Curtmola et al. [8].Our early work [22] solves secure ranked keyword search which utilizes keyword frequency to rank results instead of returning undifferentiated results. However, it only supports single keyword search. In the public key setting, Boneh etal. [9] present the first searchable encryption construction, where anyone with public key can write to the data stored on server but only authorized users with private key can search. Public key solutions are usually computationally expensive however. Furthermore, the keyword privacy could not be protected in the public key setting since server could encrypt any keyword with public key and then use the received trapdoor to evaluate this ciphertext.

B. Boolean Keyword Searchable Encryption

These schemes incur large overhead caused by their fundamental primitives, such as computation cost by bilinear map, e.g. [16], or communication cost by secret sharing, e.g. [15]. As a more general search approach, predicate encryption schemes [19]–[21] are recently proposed to support both conjunctive and disjunctive search. Conjunctive keyword search returns “all-or-nothing”, which means it only returns those documents in which all the keywords specified by the search query appear; disjunctive keyword search returns undifferentiated results, which means it returns every document that contains a subset of the specific keywords, even only one keyword of interest. In short, none of existing Boolean keyword searchable encryption schemes support multiple keywords ranked search over encrypted cloud data while preserving privacy as we propose to explore in this paper. Note that, inner product queries in predicate encryption only predicates whether two vectors are orthogonal or not, i.e., the inner product value is concealed except when it equals zero. Without providing the capability to compare concealed inner products, predicate encryption is not qualified for performing ranked search. Furthermore, most of these schemes are built upon the expensive evaluation of pairing operations on elliptic curves. Such inefficiency disadvantage also limits their practical performance when deployed in the cloud. On a different front, the research on top-*k* retrieval [27] in database community is also loosely connected to our problem.

VI. CONCLUSION

In this paper, for the first time we define and solve the problem of multi-keyword ranked search over encrypted cloud data, and establish a variety of privacy requirements. Among various multi-keyword semantics, we choose the efficient similarity measure of “coordinate matching”, i.e., as many matches as possible, to effectively capture the relevance of outsourced documents to the query keywords, and use “inner product similarity” to quantitatively evaluate such similarity measure. For meeting the challenge of supporting multi-keyword semantic without privacy breaches, we propose a basic idea of MRSE using secure inner product computation. Then we give two improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given, and experiments on the real-world dataset show our proposed schemes introduce low overhead on both computation and communication.

In our future work, we will explore supporting other multikeyword semantics (e.g., weighted query) over encrypted dataand checking the integrity of the rank order in the search result.

VII. ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation under grants CNS-0716306, CNS-0831628, CNS-0746977, and CNS- 0831963.

[1] REFERENCES

[2] [1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break

[3] in the clouds: towards a cloud definition,” ACM SIGCOMM Comput.

[4] Commun. Rev., vol. 39, no. 1, pp. 50–55, 2009.

[5] [2] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in RLCPS,

[6] January 2010, LNCS. Springer, Heidelberg.

[7] [3] A. Singhal, “Modern information retrieval: A brief overview,” IEEE

[8] Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.

[9] [4] I. H. Witten, A. Moffat, and T. C. Bell, “Managing gigabytes: Compressing and indexing documents and images,” Morgan Kaufmann Publishing,

[10] San Francisco, May 1999.

[11] [5] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of S&P, 2000.

[12] [6] E.-J. Goh, “Secure indexes,” Cryptology ePrint Archive, 2003, http:// eprint.iacr.org/2003/216.

[13] [7] Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Proc. of ACNS, 2005.

[14] [8] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,”

[15] in Proc. of ACM CCS, 2006.

[16] [9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. of EUROCRYPT, 2004.

[17] [10] M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and efficiently searchable encryption,” in Proc. of CRYPTO, 2007.

[18] [11] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, “Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions,” J. Cryptol., vol. 21, no. 3, pp. 350–391, 2008.

[19] [12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing,” in Proc. of IEEE INFOCOM’10 Mini-Conference, San Diego, CA, USA, March 2010.

[20] [13] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, “Public key encryption that allows pir queries,” in Proc. of CRYPTO, 2007.

[21] [14] P. Golle, J. Staddon, and B. Waters, “Secure conjunctive keyword search over encrypted data,” in Proc. of ACNS, 2004, pp. 31–45.

[22] [15] L. Ballard, S. Kamara, and F. Monrose, “Achieving efficient conjunctive keyword searches over encrypted data,” in Proc. of ICICS, 2005.

[23] [16] D. Boneh and B. Waters, “Conjunctive, subset, and range queries on encrypted data,” in Proc. of TCC, 2007, pp. 535–554.

[24] [17] R. Brinkman, “Searching in encrypted data,” in University of Twente, PhD thesis, 2007.

[25] [18] Y. Hwang and P. Lee, “Public key encryption with conjunctive keyword search and its extension to a multi-user system,” in Pairing, 2007.

[26] [19] J. Katz, A. Sahai, and B. Waters, “Predicate encryption supporting disjunctions, polynomial equations, and inner products,” in Proc. of EUROCRYPT, 2008.

[27] [20] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, “Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,” in Proc. of EUROCRYPT, 2010.

[28] [21] E. Shen, E. Shi, and B. Waters, “Predicate privacy in encryption systems,” in Proc. of TCC, 2009.

[29] [22] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in Proc. of ICDCS’10, 2010.