Authorized Security Preserving Multiple Keyword Ranked Search for
Encrypted Cloud Data

VENKATESWARLU KANNEKANTI M.Tech (CSE), 2K. SANATH KUMAR, 3Ravi Mathey
Associate Professor, 1 CSE Department, 2 HOD CSE Department
1,2,3 Vidya Jyothi Institute of Technology

Abstract—In cloud computing, clients usually outsource their data to the cloud storage servers to reduce the management costs. While those data may contain sensitive personal information, the cloud servers cannot be fully trusted in protecting them. Encryption is a promising way to protect the confidentiality of the outsourced data, but it also introduces much difficulty to performing effective searches over encrypted information. With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to 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 cloud, it is crucial for the search service to allow multi-keyword query and provide result similarity ranking to meet the effective data retrieval need. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely differentiate 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), and establish a set of strict privacy requirements for such a secure cloud data utilization system to become a reality. Among various multi-keyword semantics, we choose the efficient principle of “coordinate matching”, i.e., as many matches as possible, to capture the similarity between search query and data documents, and further use “inner product similarity” to quantitatively formalize such principle for similarity measurement. We first propose a basic MRSE scheme using secure inner product computation, and then significantly improve it to meet different privacy requirements in two levels of threat models. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes.

I. Introduction

Cloud Computing, the new term for the long dreamed vision of computing as a utility [1], enables convenient, on-demand network access to a centralized pool of configurable computing resources (e.g., networks, applications, and services) that can be rapidly deployed with great efficiency and minimal management overhead [2]. Cloud computing is gaining much momentum in the IT industry. Especially, we have seen the dramatic growth of public clouds, in which the computing resources can be accessed by the general public. One of the biggest advantages of a public cloud is its virtually unlimited data storage capabilities and elastic resource provisioning [3]. Many IT enterprises and individuals are outsourcing their databases to the cloud servers, in order to enjoy the much lower data management cost than maintaining their own data centers. It has never been easier than now that a variety of users/clients could access or share information stored in the cloud, independent of their locations. Despite enthusiasm around the cloud data service outsourcing model, its promises cannot be fulfilled until we address the serious security and privacy concerns that data owners have. The outsourced data may contain very sensitive information, such as Personal Health Records (PHRs), facebook photos, and business documents. Many people remain dubious about the levels of privacy protection of their data when stored in a server owned by a third-party cloud service provider. Given that there have been numerous reported data breaches related to cloud servers [2], which could be due to malicious attacks, theft or internal software bugs and errors, it can be claimed that the servers are not fully trustworthy. This implies that there is no absolute governance about how the owners’ information will be used and whether the owners actually control access to their data. To cope with the tough trust issues and to ensure owners’ control over their own privacy, applying data encryption on the documents before outsourcing has been
proposed as a promising solution, which is already adopted by many recent works [26], [7], [24]. In this paper, we focus on the “multi-owner” setting, where the encrypted data are contributed by multiple owners and can be searched by multiple users. On the one hand, to meet the effective data retrieval need, large amount of documents demand cloud server to perform result relevance ranking, instead of returning undifferentiated result. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection [3]. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the “pay-as-youuse” cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve search result accuracy as well as enhance user searching experience, it is also crucial for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse result. As a common practice indicated by today’s web search engines, data users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data. And each keyword in the search request is able to help narrow down the search result further. “Coordinate matching” [4], i.e., as many matches as possible, is an efficient principle among such multi-keyword semantics to refine the result relevance, and has been widely used in the plaintext information retrieval (IR) community.

![Fig1. Search Over Encrypted Cloud](image)

we define and solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving strict system-wise privacy in cloud computing paradigm. Among various multi-keyword semantics, we choose the efficient principle of “coordinate matching”, i.e., as many matches as possible, to capture the similarity between search query and data documents. Specifically, we use “inner product similarity” [4], i.e., the number of query keywords appearing in a document, to quantitatively evaluate the similarity of that document to the search query in “coordinate matching” principle. During index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by inner product of query vector with data vector. However, directly outsourcing data vector or query vector will violate index privacy or search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we propose a basic MRSE scheme using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique [4], and then improve it step by step to achieve various privacy requirements in two levels of threat models.

II. Related Work

Traditional searchable encryption [5–13] has been widely studied in the context of cryptography. Among those works, most are focused on efficiency improvements and security definition formalizations. The first construction of searchable encryption was proposed by Song et al. [6], in which each word in the document is encrypted independently under a special two-layered encryption construction. Goh [7] proposed to use Bloom filters to construct the indexes for the data files. For each file, a Bloom filter containing trapdoors of all unique words is built up and stored on the server. To search for a word, the user generates the search request by computing the trapdoor of the word and sends it to the server. Upon receiving the request, the server tests if any Bloom filter contains the trapdoor of the query word and returns the corresponding file identifiers. To achieve more efficient search, Chang et al. [10] and Curtmola et al. [11] both proposed similar “index” approaches, where a single encrypted hash table index is built for the entire file collection. Considering a 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 data documents $F$ to be outsourced to cloud server in the encrypted form $C$. To enable the searching capability over $C$ for effective data utilization, 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 cloud server. To search the document collection for a given keyword, an authorized user acquires a corresponding trapdoor $T$ through search control mechanisms, e.g., broadcast encryption [8]. Upon receiving $T$ from data users, cloud server is responsible to search the index $I$ and return the corresponding set of encrypted documents. To improve document retrieval accuracy, search result should be ranked by cloud server according to some ranking criteria (e.g., coordinate matching, as will be introduced shortly). Moreover, to reduce communication cost, data user may send an optional number $k$ along with the trapdoor $T$ so that cloud server only sends back top-$k$ documents that are most relevant to the search query. Finally, the access control mechanism is employed to manage decryption capabilities given to users.

**Threat Model**

Cloud server is considered as “honest-but-curious” in our model, which is consistent with the most related works on searchable encryption. Specifically, 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 cloud server knows, we consider two levels of threat models as follows. Known Ciphertext Model In this model, cloud server is supposed to only know encrypted dataset $C$ and searchable index $I$, both of which are outsourced from data owner. Known Background Model In this stronger model, cloud server is supposed to possess some backgrounds on the dataset, such as the subject and its related statistical information, in addition to what can be accessed in known ciphertext model. As an instance of possible attacks in this case, cloud server could utilize document frequency or keyword frequency [23] to identify keywords in the query. As a hybrid of conjunctive search and disjunctive search, “coordinate matching” [4] is an intermediate approach which uses the number of query keywords appearing in the document to quantify the similarity of that document to the query. When users know the exact subset of the dataset to be retrieved, Boolean queries perform well with the precise search requirement specified by the user. In cloud computing, however, this is not the practical case, given the huge amount of outsourced data. Therefore, it is more flexible for users to specify a list of keywords indicating their interest and retrieve the most relevant documents with rank order. For the threat model, we assume that the proxy server(s) are semi-trusted, and an adversary cannot control both cloud server and proxy server(s) at the same time. We argue this is reasonable in practice since a proxy server can be well protected by the organization that owns it (e.g., the TA’s organization). Also, we assume the cloud server do not launch active attacks such as probe-response attack [8] which needs a large amount of partial ciphertexts be sent to the proxy, as there exist some detection mechanism (e.g., traffic monitoring). We depict the enhanced system framework in. The TA generates a random secret $r$ in addition to $MSK$ and $PK$, the proxy is given $r$, which is also embedded into the capabilities for the LTAs and users.

**III. Privacy Requirements of MRSE**

We define the framework of multi-keyword ranked search over encrypted cloud data (MRSE) and establish various strict system-wise privacy requirements for such a secure cloud data utilization system. For easy presentation, operations on the data documents are not shown in the framework since data owner could easily employ traditional symmetric key cryptography to encrypt and then outsource data. With focus on index and query, a MRSE consists of four algorithms as follows. ² Setup(1) Taking a security parameter $\lambda$ as input, data owner outputs a symmetric key as $SK$. ² BuildIndex($F$; $SK$) Based on the dataset $F$, data owner builds a searchable index $I$ which is encrypted by the symmetric key $SK$ and then outsourced to cloud server. After the index construction, the document collection can be independently encrypted and outsourced. ² Trapdoor($F$/$W$) With $t$ keywords of interest in $F$ as input, this algorithm generates a corresponding trapdoor $T$ of $W$. ² Query($T$/$W$; $K$; $I$) When cloud server receives a query request as $(T$/$W$, $k$), it performs the ranked search on the index $I$ with the help of trapdoor $T$/$W$, and
finally returns F e W, the ranked id list of top-k documents sorted by their similarity with F W. Both search control and access control are not 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. The representative privacy guarantee in the related literature, such as searchable encryption, is that the server should learn nothing but search results. With this general privacy description, we explore and establish a set of strict privacy requirements specifically for the MRSE framework. As for the data privacy, data owner can resort to traditional symmetric key cryptography to encrypt the data before outsourcing, and successfully prevent cloud server from prying into outsourced data. With respect to the index privacy, if 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 [23]. Therefore, searchable index should be constructed to prevent server from performing such kind of association attack. While data and index privacy guarantees are demanded by default in the related literature, various search privacy requirements involved in the query procedure are more complex and difficult to tackle as follows. Keyword Privacy As users usually prefer to keep their search from being exposed to others like 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, 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 [24]. When cloud server knows some background information of the dataset, this keyword specific information may be utilized to reverse engineer the keyword. Trapdoor Privacy Since only authorized users are allowed to acquire trapdoors for their search query, the server is not expected to have the ability to generate valid trapdoors from previous received ones. Specifically, given one trapdoor for a set of multiple keywords, the server is not allowed to generate a valid trapdoor for its subset, including single keyword.

IV. Performance Evolution

We conducted a thorough experimental evaluation of the proposed techniques on real data we demonstrate a thorough experimental evaluation of the proposed technique on a real-world dataset the Enron Email Dataset [27]. We randomly select different number of emails to build dataset. The whole experiment system is implemented by C language on a Linux Server with Intel Xeon Processor 2.93GHz. The public utility routines by Numerical Recipes are employed to compute the inverse of matrix. The performance of our technique is evaluated regarding the efficiency of three proposed MRSE schemes, as well as the tradeoff between search precision and privacy. As presented in 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 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. To evaluate the purity of the k documents retrieved by user, we define a measure as precision Pk = k0=k where k0 is number of real top-k relevant documents that are returned by cloud server. Fig. 2 shows that the precision in MRSE is evidently affected by the standard deviation ¾ of the random variable "x. From the consideration of effectiveness, standard deviation is expected to be smaller so as to obtain high precision indicating the good purity of retrieved documents. However, user privacy may have been partially leaked to cloud server as a consequence of small Access pattern is defined as the sequence of ranked search results. Although search results cannot be protected (excluding costly PIR technique), we can still hide the rank order of retrieved documents as much as possible. In order to evaluate this privacy guarantee, we first define the rank perturbation as epi = rRI r0 RI, where ri is the rank number of document i in the
retrieved top-k documents and r_v_i is its rank number in the real top-i ranked documents. The overall rank we can see that small leads to higher precision of search result but lower rank privacy guarantee, while large results in higher rank privacy guarantee but lower precision. In other words, our scheme provides a balance parameter for data users to satisfy their different requirements on precision and rank privacy.

V. Conclusion

In this paper, for the first time we formalize and solve the problem of supporting efficient yet privacy-preserving 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 principle of “coordinate matching”, i.e., as many matches as possible, to effectively capture similarity between query keywords and outsourced documents, and use “inner product similarity” to quantitatively formalize such a principle for similarity measurement. For meeting the challenge of supporting multi-keyword semantic without privacy breaches, we first propose a basic MRSE scheme using secure inner product computation, and significantly improve it to achieve privacy requirements in two levels of 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. As our future work, we will explore supporting other multi-keyword semantics (e.g., weighted query) over encrypted data, integrity check of rank order in search result and privacy guarantees in more stronger threat model.

References


Ravi Mathey is a post-graduate specialized in Computer Science from BIT -Ranchi and he did Instrumentation technology in Andhra University. He has more than 20 years of experience in Research and Development; presently he is working as a Associate Professor and HOD of CSE Department at Vidya Jyothi Institute of Technology (VJIT). His area of research is wireless embedded application, cloud computing, image compression techniques by using fractals and wavelets.


K. Sanath Kumar Working as Associate Professor in Vidya Jyothi