

# Efficient Information Retrieval in Cost-Effective Cloud Environment with Privacy Preserving

Abdul Khader<sup>1</sup>, Henin Karkeda<sup>2</sup>

<sup>1</sup>M.Tech Student, Department of Computer Science and Engineering  
KLE Dr. M. S. Sheshgiri College of Engineering and Technology  
Belgaum, Karnataka, India

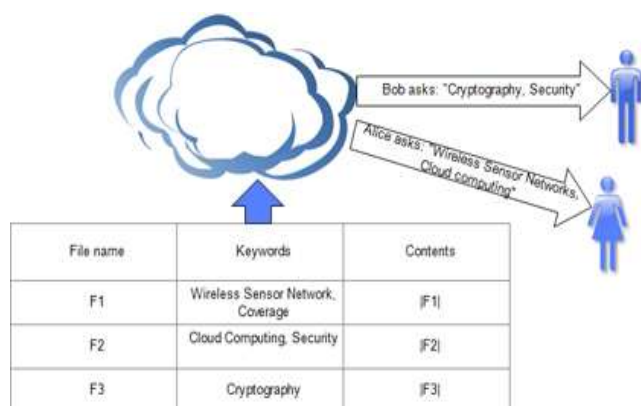
<sup>2</sup>M.Tech Student, Department of computer science and engineering  
Visvesvaraya Technological University  
Belgaum, Karnataka, India

**Abstract:** Cloud computing as an emerging technology trend is expected to reshape the advances in information technology. With the increasing popularity of cloud computing there is increased motivation to outsource data services to the cloud to save money. An important problem in cloud environment is to protect user's privacy while querying data from the cloud. The researchers have proposed several techniques to address this problem. However, existing technologies incur heavy computational cost and bandwidth related cost. In this paper we propose an aggregation and distribution layer (ADL), we present scheme, termed efficient information retrieval for ranked query (EIRQ), to further reduce querying costs incurred in the cloud. All the queries will be grouped into multiple ranks, such that a higher ranked query can retrieve a higher percentage of matched files. The users are allowed to enter the file name also in case if he knows the exact file to be retrieved. This will help to stop the retrieval of unnecessary files that would have been fetched if only keywords were used.

**Keywords:** Cloud computing, communication cost, privacy, computational cost

## 1. Introduction

Due to the overwhelming merits of cloud computing, such as scalability cost-effectiveness, and flexibility, more and more organizations are willing to outsource their data for storing in the cloud. The benefits of utilizing the cloud [2] (lower operating costs, elasticity and so on) come with a trade-off. Users will have to entrust their data to a potentially untrustworthy cloud provider. As a result, cloud security has become an important problem for both industry and academia. One important security problem is the potential privacy leakages that may occur when outsourcing data to the cloud. For instance, let us consider the application scenario as shown in Fig. 1



**Figure 1:** Application scenario

Files F1, F2, and F3 stored in the cloud are described with keywords "Wireless sensor network, Coverage", "Cloud Computing, Security", and "Cryptography", respectively. Alice uses keywords "Wireless sensor networks, Cloud Computing", and Bob uses keywords "Cryptography, Security" to query data from cloud.

When the users want to search for some files, they will send a query to the cloud with certain keywords. The cloud will evaluate the query and return the necessary files to the users. During this process, the cloud will know what files the user is interested in from observing the query and the type of the files returned to that user. Preventing a leak of this type of information to the cloud is difficult since the cloud must have access to the information to efficiently return the appropriate files to the users.

An organization subscribes the cloud services and authorizes its staff to share files in the cloud. Each file is described by a set of keywords, and the authorized users, can retrieve files of their interests by querying the cloud with relevant keywords. Since a cloud is operated by a third party, there have been some concerns over the possible privacy leaks that may occur. Such concerns have led researchers to propose various techniques to protect user privacy. Alternatively, if we can combine more than one query together, we can save the overhead by reducing the number of queries that the server has to process.

A key privacy search solution was proposed by Ostrovskiy et al. [1] that provides the same privacy level as downloading the entire database from the cloud with significantly less communication costs. The cloud cannot know which files are really interested by a user by asking the cloud to return the entire database. However, the Ostrovsky scheme has a high computation cost, since it must require the cloud to process the encrypted query on every file in a collection.

To make private search applicable in a cloud environment we propose a system that reduces the computational cost and communication costs while providing similar privacy protection as in the prior protocols. Our solution introduces a proxy server called aggregation and distribution layer (ADL)

– a layer in between the cloud and the users. The users will send the queries to the ADL first instead of cloud directly and this ADL will query the cloud on behalf of users. Thus the cloud needs to execute the aggregated query only once to return files matching all queries to the ADL. Under the ADL, the computation cost incurred on the cloud can be largely reduced, since the cloud only needs to execute a combined query once, no matter how many users are executing queries. Furthermore, the communication cost incurred on the cloud will also be reduced, since files shared by the users need to be returned only once. Most importantly, by using a series of secure functions, COPS can protect user privacy from the ADL, the cloud, and other users.

The users are also allowed to decide personally how many matched files he wants to be returned. This is motivated by the fact that in some cases, there are a lot of files matching a user's query, but the user is interested in only a certain percentage of matched files. To illustrate, let us assume that Alice wants to retrieve 4% of the files that contain keywords

A, B, and Bob wants to retrieve 40% of the files that contain keywords A, C. The cloud holds 1,000 files, where  $\{F1, \dots, F500\}$  and  $\{F501, \dots, F1000\}$  are described by keywords A, B and A, C, respectively. In the Ostrovsky scheme, the cloud will have to return 4,000 files. In the COPS scheme, the cloud will have to return 2,000 files. In our scheme, the cloud only needs to return 400 files. The bandwidth consumed in the cloud can be largely reduced by allowing the users to retrieve matched files according to their demand.

We propose a scheme, termed Efficient Information retrieval for Ranked Query (EIRQ), in which each user can choose the rank of his query to determine the percentage of matched files to be returned. The idea of EIRQ is to construct a mask matrix that helps in privacy preserving which allows the cloud to filter out a certain percentage of matched files before returning to the ADL. The cloud should correctly filter out files according to the rank of queries without knowing anything about user privacy.

## 2. Related Work

Our work is on protecting user privacy while searching data on untrusted servers. User privacy can be classified into search privacy and access privacy [3]. Search privacy means that the servers know nothing about what the user are searching for, and access privacy means that the cloud knows nothing about which files are returned to the user. There has been a lot of work conducted in this field including private searching [4, 5, 6, 7], private retrieval information (PIR) [8], and searchable encryption, where user privacy can be protected in private searching and PIR, but only search privacy can be protected in private searching and PIR, but only search privacy can be protected in searchable encryption.

Private searching was first proposed by [2], where data is stored in the clear form, and the query is encrypted with the Paillier cryptosystem [14] that exhibits the homomorphic properties. Ranked searchable encryption enables users to retrieve the most matched files from the cloud in the case that both the query and data are in the encrypted form. The work

by [9], which only supports single-keyword searches, encrypts files and queries with Order Preserving Symmetric Encryption (OPSE) [10] and utilizes keyword frequency to rank results. Their following work [11], which supports multiple-keyword searches, uses the secure KNN technique [12] to rank results based on inner products. The main limitation of these approaches is that user access privacy [13] will not be preserved.

## 3. Background

### A. System Model

The system consists of three types of entities: cloud, aggregation and distribution layer (ADL), and users as shown in Fig.2. For ease of explanation, in this paper, we only use a single ADL, but multiple ADLs can be deployed as necessary. Multiple files are stored in a potentially untrusted cloud, where each file is described by several distinct keywords.

The staff members, as the authorized users, send their queries to the ADL, which will aggregate user queries and send a combined query to the cloud. Then, the cloud processes the combined query on the file collection and returns a buffer that contains all of matched files to the ADL, which will distribute the search results to each user. To aggregate sufficient queries, the organization may require the ADL to wait for a period of time before running our schemes, which may incur a certain querying delay. In the supplementary file, we will discuss the computation and communication costs as well as the querying delay incurred on the ADL.

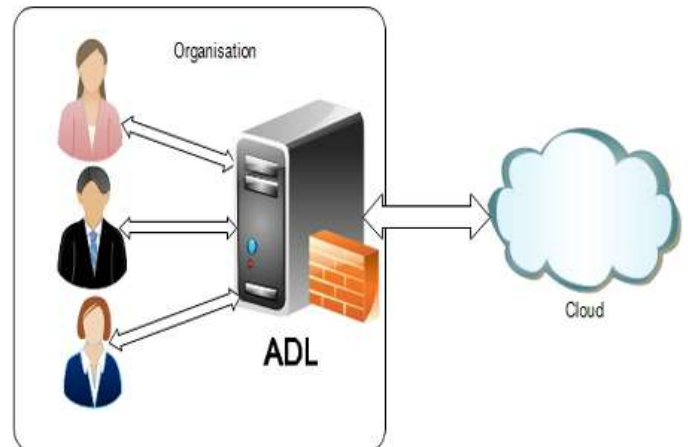


Figure 2: System model

### B. Security Requirements

The ADL is assumed to be trusted by all of the users since it is deployed within the organisation itself, and the communication channels are assumed to be secured under security protocols like SSL. Each user individually sends the query to the ADL, which will distribute appropriate files to each user. As long as the ADL is trusted and correctly executes our schemes, the user cannot know anything about other users' interests. Thus, the cloud is the only adversary for each user. The cloud is assumed to be honest but curious. That is, it will obey our schemes, but still want to know some additional information. User privacy can be divided into search privacy and access privacy, where the cloud neither

learns what the user is searching for nor is which files returned to a user. Since user queries are classified into multiple ranks, rank privacy, a new kind of user privacy, also needs to be protected against the cloud. Rank privacy entails hiding the rank of each query from the cloud, i.e., the cloud provides differential query services without knowing which level of service is chosen by the user.

Our security goal is to thoroughly protect user search privacy, access privacy, and rank privacy against the cloud. All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

#### 4. Protocol Description

##### A. Intuition

The basic idea of EIQR is that a privacy-preserving mask matrix is used to filter out a certain percentage of files before mapping them to a buffer. Before illustrating EIQR, two fundamental problems should be resolved:

First, we should determine the relationship between query rank and the percentage of returned matched files. Suppose that queries are classified into  $r$  ranks, where Rank-0 queries have the highest rank and Rank- $r$  queries have the lowest rank. Rank-0 queries can retrieve 100% of the matched files, and Rank- $r$  queries cannot retrieve any files.

Second, we should determine which matched files will be returned and which will not. In this paper, we simply determine the probability of a file being returned by the highest rank of queries matching this file.

##### B. EIRQ

EIRQ consists of four algorithms, as shown in Fig. 4. We will use the following example to describe its working process. The dictionary and files are the same as in Section III-(C); users are classified into four ranks, where Alice, a Rank-0 user, queries with keywords “A, B”, and Bob, a Rank-1 user, uses keywords “A, C”. According to our rules, “A, B” is Rank-0 keywords, “C” is a Rank-1 keyword, and “D” is a Rank-4 keyword. Correspondingly, F1 and F2 are Rank-0 files which will be returned with a probability of 1, F3 and F4 are Rank-1 files which will be returned with a probability of 75%, and F5 is a Rank-4 file which will not be returned.

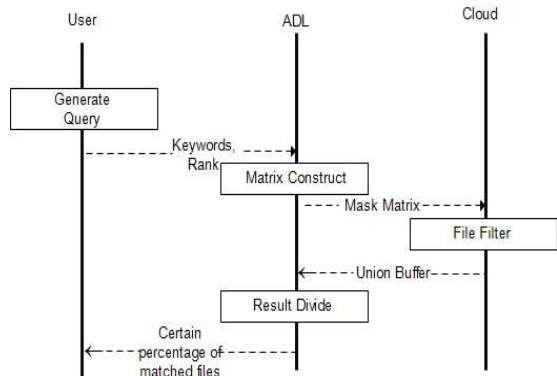


Figure 3: Working process of EIRQ

Step 1: Each user runs the sends the query to the ADL, where the user query consists of the chosen keywords and the query rank.

Step 2: Given users queries, the ADL runs the Matrix-Construct algorithm (Alg. 1) to send a mask matrix to the Cloud.

Step 3: Based on the mask matrix, the cloud runs the FileFilter algorithm (Alg. 2) to filter out a certain percentage of matched files and returns a union buffer to the ADL.

Step 4: The ADL runs the Result Divide algorithm to distribute files to each user. The ADL first recovers all files that match user queries as the File Recover algorithm.

```

for i
= 1 to d do Set l to be the highest query rank choosing t
keyword in Dic
    for j = 1 to r do
        if l + j ≤ r then M[i, j] = 1 else M[i, j]
= 0 Encrypt M[i, j] with the ADL's public key
    
```

Alg. 1: Matrix Construct

```

for each file Fj stored in the cloud do
    for i = 1 to d do
        k = j mod r; cj = πDic[i] ∈ Fj M[i, k]; ej = cj|Fj|
    Multiply pair (cj, ej) many times to a compact buffer
    
```

Alg.2: File Filter

#### 5. Security Analysis

We will show that EIRQ can provide search privacy, access privacy, and rank privacy as follows:

- 1) Search privacy: In EIRQ, the combined query (the mask matrix) from the ADL to the cloud is encrypted with the ADL’s public key. Therefore, the cloud cannot deduce what each user is searching for from the encrypted query.
- 2) Access privacy: In EIRQ, the cloud processes each file similarly to generate a compact buffer where unmatched files are encrypted to 0, while conducting searches. The buffer returned to the ADL is encrypted with the ADL’s public key. Therefore, the cloud cannot know which files are actually returned from the encrypted buffer.
- 3) Rank privacy: In EIRQ, the mask matrix from the ADL to the cloud is a  $d$ -row and  $r$ -column matrix, where  $r$  is the information that is the information that we leak more than [1]. Given  $r$ , the cloud only knows that all users are classified into ranks without knowing how many users are in each rank, nor which users are in which ranks. Therefore, user rank privacy is protected.

#### 6. Conclusion

In this paper, we proposed three EIRQ schemes based on an ADL to provide differential query services while protecting user privacy. By using our schemes, a user can retrieve different percentages of matched files by specifying queries of different ranks. By further reducing the communication cost incurred on the cloud, the EIRQ schemes make the private searching technique more applicable to a cost-efficient cloud environment. However, in the EIRQ schemes, we

simply determine the rank of each file by the highest rank of queries it matches. For our future work, we will try to design a flexible ranking mechanism for the EIRQ schemes

## References

- [1] R. Ostrovsky and W. Skeith III, "Private searching on streaming data," in Proc. of ACM CRYPTO, 2005.
- [2] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," NIST Special Publication, 2011.
- [3] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS, 2006.
- [4] J. Bethencourt, D. Song, and B. Waters, "New constructions and practical applications for private stream searching," in Proc. Of IEEE S&P, 2006.
- [5] G. Danezis and C. Diaz, "Improving the decoding efficiency of private search," in IACR Eprint archive number 024, 2006.
- [6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval," Journal of ACM , 1995.
- [7] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of IEEE ICDCS, 2010.
- [8] A. Boldyreva, N. Chenette, Y. Lee, and A. Oneill, "Order-preserving symmetric encryption," Advances in Cryptology-EUROCRYPT, 2009.
- [9] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi keyword ranked search over encrypted cloud data," in Proc. of IEEE INFOCOM, 2011.
- [10] W. Wong, D. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in Proc. of ACM SIGMOD, 2009.
- [11] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS, 2006.
- [12] I. Damagard and M. Jurik, "A generalization, a Simplification and Some Applications of Pailler's Probabilistic Public Key System" in proceedings of PKC, 2001.

## Author Profile



**Abdul Khader** received the Diploma in Computer Science and Engineering from SNMP Moodbidri and B.E. in Computer Science and Engineering from MITE Mangalore and currently pursuing Masters in Computer Science and Engineering from KLE Dr. M. S. Sheshgiri college of Engineering and Technology, Belgaum.



**Henin Roland Karkadah** has been awarded Diploma in Computer Science & Engineering from T.M.A PAI Polytechnic Manipal, Graduated from MITE Mangalore ( B.E. in Computer Science and Engineering) and he is currently pursuing MTech in Computer Science and Engineering at Visvesvaraya Technological University (VTU), Belgaum.