EIRQ Methods to Provide a Cost-Efficient Solution for Private Searching in Cloud Computing

[1]G.Satya Suneetha, [2]S.V.Ramana Murthy, [3]T.S.V.V.S.Savithri
[1,2,3]Dept.of CSE, Pragati Engineering College, Surampalem, EGDT, AP, India

Abstract
As a characteristic cloud application an organization pledge the cloud services and approves its team to share files in the cloud. Each file is explained by a set of keywords and the staff as authorized users can repossess files of their interests by querying the cloud with certain keywords. In such an environment how to protect user privacy from the cloud which is a third party outside the security boundary of the organization turn into a key problem. The communication cost acquires on the cloud will also be concentrated since files shared by the users need to be returned only once. Most significantly by using a series of secure functions COPS can protect user privacy from the ADL the cloud and other users. The main drawback is that it will cause a heavy querying overhead incurred on the cloud and thus goes against the original intention of cost efficiency. In this paper we present a method termed efficient information retrieval for ranked query (EIRQ) based on an aggregation and distribution layer (ADL) to condense querying overhead deserved on the cloud.

Keywords
Cloud Computing, Cost Efficiency, Differential Query Services, Privacy

I. Introduction

User privacy can be classified into search privacy and access privacy. Search privacy means that the cloud knows nothing about what the user is searching for and access privacy means that the cloud knows nothing about which files are returned to the user. When the files are stored in the clear forms a immature solution to protect user privacy is for the user to request all of the files from the cloud. This way the cloud cannot know which files the user is really interested in. While this does provide the necessary privacy and the communication cost is high. The ADL deployed inside an organization has two main functionalities, aggregating user queries and distributing search results. Under the ADL the computation cost incurred on the cloud can be basically condensed since the cloud only needs to complete a combined query once no matter how many users are executing queries. Under different parameter settings, extensive evaluations have been conducted on both analytical models and on a real cloud environment, in order to examine the effectiveness of our schemes. In EIRQ queries are classified into multiple ranks where a higher ranked query can regain a senior percentage of matched files. A user can retrieve files on demand by choosing queries of different ranks.

II. Related Work

Private searching performs keyword-based searches on unencrypted data. Private searching was first proposed which let a server to filter streaming data without cooperation of user privacy. Their solution requires the server to return a buffer of size O(f log(f)) when f files match a user's query. Each file is associated with a survival rate which denotes the probability of this file being productively recovered by the user. Based on the Paillier cryptosystem the files that disparity a query will not stay alive in the buffer but the matched files take pleasure in a high survival rate. When applying these schemes to a large-scale cloud environment querying costs will be widespread.

III. Existing Method

Confidential searching allows a user to get back files of interest from an untrusted server without escape of any information. However the system has a high computational cost since it requires the cloud to process the query on every file in a collection. Otherwise the cloud will learn that certain files without processing are of no attention to the user. It will quickly become a presentation bottleneck when the cloud needs to procedure thousands of queries over a collection of hundreds of thousands of files.

IV. Disadvantages

The main problem is that it will cause a serious querying overhead incurred on the cloud and thus goes against the original intention of cost efficiency.

V. Proposed Method

The essential idea of EIRQ is to make a privacy preserving mask matrix that allows the cloud to filter out a certain percentage of matched files before returning to the ADL. Since the cloud needs to correctly filter out files according to the rank of queries without knowing anything about user privacy.

VI. Advantages

A user can get back files on claim by choosing queries of different ranks. This feature is useful when there are a large number of matched files but the user only needs a small subset of them.

VII. System Architecture

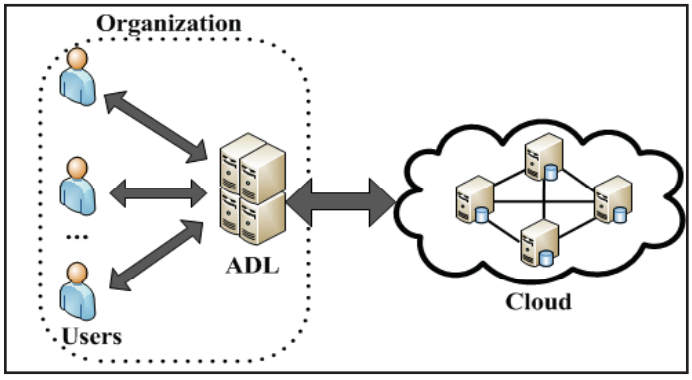

Fig. 1:

VIII. Differential Query Services

We bring in a novel concept differential query services to COPS where the users are allowed to make a decision how many matched files will be returned. This is inspired by the fact that under certain 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. In the COPS scheme the cloud will have to return 1, 000 files. In our scheme the cloud only needs to return 200 files. Consequently by allowing the users to get back matched files on demand the bandwidth consumed in the cloud can be mainly abridged.

VIII. Information Retrieval For Ranked Query

In Efficient Information retrieval for Ranked Query (EIRQ) where each user can choose the rank of his query to decide the percentage of matched files to be returned. The basic idea of EIRQ is to construct a privacy preserving mask matrix that allows the cloud to sieve out a certain percentage of matched files before returning to the ADL. This is not a trivial work since the cloud needs to correctly filter out files according to the rank of queries without significant anything about user privacy. Focusing on different design goals we provide two extensions, the first extension highlights simplicity by requiring the least amount of modifications from the Ostrovsky scheme and the second extension emphasizes privacy by leaking the least amount of information to the cloud.

IX. Aggregation and Distribution Layer

An ADL is arranged in an organization that approves its staff to share data in the cloud. The staff members, as the authorized users send their queries to the ADL which will collect user queries and send a combined query to the cloud. Then the cloud processes the mutual 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 collective sufficient queries the organization may need the ADL to wait for a period of time before running our schemes which may invite a certain querying delay.

X. Ranked Queries

To further decrease communication cost a disparity query service is provided by permitting each user to get back matched files on demand. Especially a user selects a particular rank for his query to decide the percentage of matched files to be returned. This feature is useful when there are a lot of files that match a user's query but the user only need a small subset of them.

XI. Process Flow

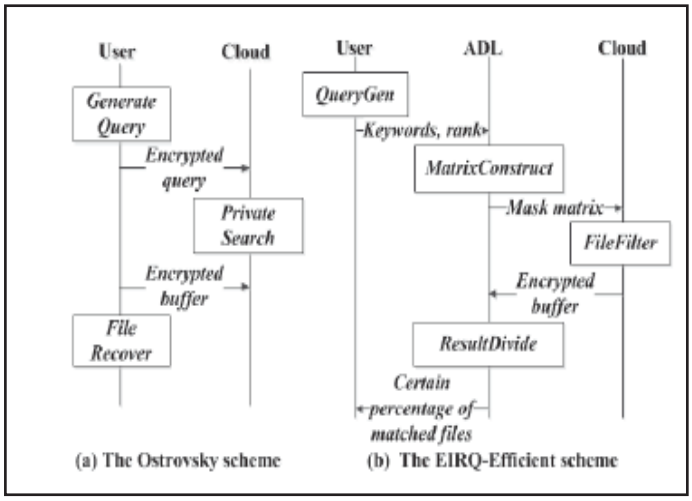

Fig. 2: Working Process

XII. Algorithms Used

We have 3 EIRQ schemes,

1. EIRQ-Efficient Scheme
2. EIRQ-Simple Scheme
3. EIRQ-Privacy Scheme

XIII. EIRQ-Efficient Scheme

First we should establish the relationship between query rank and percentage of matched files returned.

Suppose that queries are classified into 0 ~ r ranks. Rank-0 queries have the highest rank and Rank-r queries have the lowest rank. In this paper we basically determine this relationship by allowing Rank-i queries to retrieve (1 − i/r) percent of matched files. Therefore Rank-0 queries can retrieve 100% of matched files and Rank-r queries cannot retrieve any files.
Secondly we should conclude which files will be returned and which will not. We simply resolve the probability of a file being returned by the highest rank of queries matching this file.
We simply determine the probability of a file being returned by the highest rank of queries matching this file. Specifically we first rank each keyword by the highest rank of queries choosing it and then rank each file by the highest ranks of its keywords. If the file rank is i, then the probability of being filtered out is i/r. Therefore, Rank-0 files will be mapped into a buffer with probability 1 and Rank-r files will not be mapped at all. Since unneeded files have been filtered out before mapping, the mapped files should survive in the buffer with probability 1.

The user runs the QueryGen algorithm to send keywords and the rank of the query to the ADL. Since the ADL is supposed to be a trusted third party, this query will be sent without encryption.
After aggregating enough user queries the ADL runs the Matrix Construct algorithm to send a mask matrix to the cloud.
The ADL runs the Result Divide algorithm to hand out search results to each user. File contents are recovered as the File Recover algorithm in the Ostrovsky scheme.

Algorithm 1 The EIRQ-Efficient scheme

MatrixConstruct (run by the ADL with public key pk)
for $i = 1$ to d do
  set l to be the highest rank of queries choosing $Dic[i]$
  for $j = 1$ to r do
    if $j \leq r - l$ then
      $M[i, j] = E_{pk}(1)$
    else
      $M[i, j] = E_{pk}(0)$
adjust γ and β so that file survival rate is 1
FileFilter (run by the cloud)
for each file F_j stored in the cloud do
  for $i = 1$ to d do
    $k = j \mod r$; $e_j = \prod_{Dic[i] \in F_j} M[i, k]$; $e_j = c_j^{|F_j|}$
    map (c_j, e_j) γ times to a buffer of size β

Initially we have to decide the relationship between query rank and the percentage of matched files to be returned. Secondly we should establish which matched files will be revisiting and which will not. In this paper we merely conclude the possibility of a file being returned by the highest rank of queries matching this file. Particularly we first rank each keyword by the highest rank of queries choosing it and then rank each file by the highest rank of its keywords. EIRQ primarily consists of four algorithms with its working procedure. Since algorithms QueryGen and ResultDivide are easily understood we only provide the details of algorithms MatrixConstruct and FileFilter.

XIV. Experimental Results

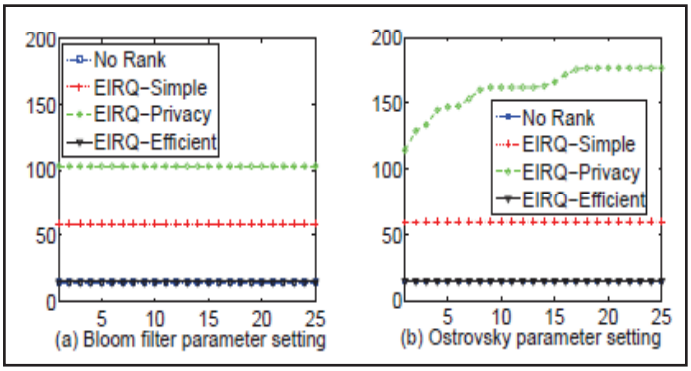

Fig. 3:

The evaluations of computational cost on the cloud are shown where the number of queries in each rank ranges from 1 to 25. Under the Bloom filter parameter setting the computational cost is approximately 14.807s in No Rank 59.274s in EIRQ. Simple 101.075s in EIRQ-Privacy and 14.861s in EIRQ Efficient. Under the Ostrovsky parameter setting the computational cost approximately ranges from 14.8270s to 14.8788s in No Rank from 59.1671s to 59.3838s in EIRQ-Simple from 114.0475s to 176.5107s in EIRQ-Privacy and from 14.8664s to 14.9269s in EIRQ Efficient. In both settings EIRQ-Privacy consumes the most computation cost and EIRQ-Efficient like No Rank consumes the smallest amount calculation charge.

XV. Future Work

It facilitates partners to make available services that use the Cloud infrastructure. The Solution EES program is intended to help Partners augment their cloud contribution and widen experience to new customers and global markets. EES offers software, applications and cloud services on top of public Cloud. The program consists of a set of tools such as platform as a service (PaaS) and software as a service (SaaS) vendors provide cloud based services include Application, Database, Development & Testing, Management e.g. Orchestration, Mobile computing, Monitoring, Multimedia, Platform as a Service, Security, Storage and Technology.

XVI. Conclusion

We planned three EIRQ schemes based on an ADL to provide differential query services while defending user privacy, a user can get back different percentages of matched files by identifying queries of different ranks. By additionally reducing the communication cost acquired on the cloud. The EIRQ schemes make the private searching method more appropriate to a cost-efficient cloud environment. However in the EIRQ schemes we just conclude the rank of each file by the highest rank of queries it matches. For our future work we will attempt to intend a supple ranking mechanism for the EIRQ schemes.

References

[1] P. Mell, T. Grance,"The nist definition of cloud computing (draft)", NIST Special Publication, 2011.
[2] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions", In Proc. of ACM CCS, 2006.
[3] R. Ostrovsky, W. Skeith,"Private searching on streaming data", In Proc. of CRYPTO, 2005.
[4] "Private searching on streaming data", Journal of Cryptology, 2007.
[5] J. Bethencourt, D. Song, B. Waters,"New constructions and practical applications for private stream searching", In Proc. Of IEEE S&P, 2006.
[6] "New techniques for private stream searching", ACM Transactions on Information and System Security, 2009.
[7] Q. Liu, C. Tan, J. Wu, G. Wang,"Cooperative private searching in clouds", Journal of Parallel and Distributed Computing, 2012.
[8] G. Danezis, C. Diaz,"Improving the decoding efficiency of private search", In IACR Eprint archive number 024, 2006.
[9] "Space-efficient private search with applications to rateless codes", Financial Cryptography and Data Security, 2007.
[10] M. Finiasz, K. Ramchandran,"Private stream search at the same communication cost as a regular search: Role of ldpc codes", In Proc. of IEEE ISIT, 2012.
[11] X. Yi, E. Bertino,"Private searching for single and conjunctive keywords on streaming data", In Proc. of ACM Workshop on Privacy in the Electronic Society, 2011.
[12] B. Hore, E.-C. Chang, M. H. Diallo, S. Mehrotra, "Indexing encrypted documents for supporting efficient keyword search", in Secure Data Management, 2012.
[13] P. Paillier,"Public-key cryptosystems based on composite degree residuosity classes", In Proc. of EUROCRYPT, 1999.
[14] Q. Liu, C. C. Tan, J. Wu, G. Wang,"Efficient information retrieval for ranked queries in cost-effective cloud environments", in Proc. of IEEE INFOCOM, 2012.
[15] S. Yu, C. Wang, K. Ren, W. Lou,"Achieving secure, scalable, and fine-grained data access control in cloud computing", in Proc. of IEEE INFOCOM, 2010.

G.Satya Suneetha, M.Tech, Assistent Professor, Dept.of CSE, Pragati Engineering College, Surampalem, EGDT, AP, India

S.V.Ramana Murthy, Dean R&D, Senior Member-IEEE, Pragati Engineering College, Surampalem, EGDT, AP, India.

T.S.V.V.S.Savithri, M.Tech student, Pragati Engineering College, Surampalem, EGDT, AP, India.