International Journal of Technology and Engineering System (IJTES)
Vol 7. No.3 2015 Pp. 250-255

available at : www.ijcns.com
ISSN: 0976-1345

EFFICIENT INFORMATION RETRIEVAL FOR RANKED QUERY SERVICES WITH COST EFFICIENT CLOUDS

T. RAJALAKSHMI[1] ,R. SEETHALAKSHMI[2]
MS. PRAGATHESWARI[3] , MS. Z. JASIMAJASMI4
[1,3,4]*UG Scholar of Department of Computer Science & Engg*
MRK Institute of Technology,
Kattumannarkoil
Raji.t029@gmail.com, pragacse1994@gmail.com

ABSTRACT

Cloud Computing is an emerging technology trend is expected to reshape the advances in information technology. In a cost-efficient cloud environment, a user can tolerate a certain degree of delay while retrieving information from the cloud to reduce costs.Existing system is private keyword based file retrieval scheme that was originally proposed by Ostrovsky. A scheme, termed Efficient Information retrieval for Ranked Query (EIRQ), based on aggregation and distribution layer, to reduce querying overhead incurred on the cloud. In EIRQ, queries are classified into multiple ranks, where a higher ranked query can retrieve a higher percentage of matched files. A user can retrieve files on demand by choosing queries of different ranks. To evaluations have been conducted on analytical models and on a real cloud environment.

Index terms: Cloud computing, cost efficiency, differential query services, privacy.

1.INTRODUCTION

Cloud computing is the use of computing resources(hardware and software) that are delivered as a service over a network.(typically the internet).The name comes from the common use of a cloud shaped symbol as an abstraction for the complex infrastructure it contains in advanced software applications and high-end networks of server computers[6] . The goal of cloud computing is to apply traditional supercomputing, or high performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them[6]. This shared IT infrastructure contains large pools of systems that are linked together. Often, Virtualization techniques are used to maximize the power of cloud computing. Cloud system diagram. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as a managed third-party services. These services typically provide access to

computing comprises three different service models, namely Infrastructure-as-a-Service(IaaS), Platform-as-a-Service(PaaS), and Software-as-a-Service(SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider. The benefits of cloud computing are Achieve economies of scale, Reduce spending on technology infrastructure, Globalize your workforce on the cheap, Streamline processes,

Reduce capital costs, Improve accessibility, monitor projects more effectively, Less personnel training is needed, Minimize licensing new software and Improve flexibility. Cloud computing refers to applications and services offered over the internet. These services are offered from data centers all over the world, which collectively are refers to us the cloud. This metaphor represents the intangible, yet universal nature of the internet.

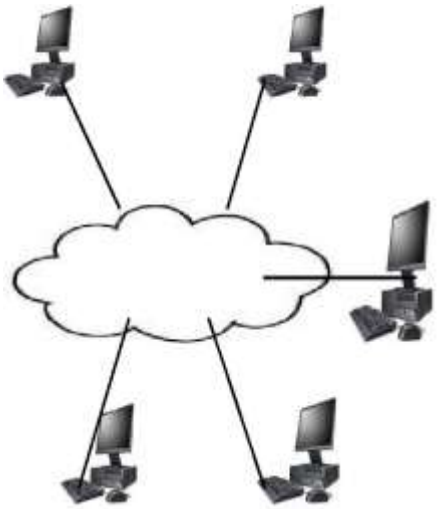

Fig. Cloud Computing

2. LITERATURE REVIEW

We introduce a model for searchable symmetric encryption(SSE) allows a party to outsource the storage of his data to another party in a private manner,while maintaining the ability to selectively search over it.This problem has been the focus of active research and several security definitions and constructions have been proposed.In this project, we begin by reviewing existing notions of security and propose new and stronger security definitions.We present two constructions that we show secure under our new definitions.Interestingly,in addition to satisfying stronger security guarantees,our construction are more efficient than all previous constructions.Further,prior work on SSE only considered the setting where only the owner of the data is capable of submitting search queries.We consider the natural extension where an arbitrary group of parties other than the owner can submit search queries.

We formally SSE in multi-user setting and present an efficient construction.A system for private stream searching allows a clients to retrieve document matching some criteria from a remote server while the server evaluating the request remains provably oblivious to the search criteria.In this extended abstract,we give a high level outline of a new scheme for this problem and an experimental analysis of its scalability.The new scheme is highly Efficient in practice. We demonstrate the pratical applicablility of the scheme by considering its performance in the demanding scenario of providing a privacy preserving version of the Google News Alerts service. We consider the problem of private searching on streaming data.We show that in this model we can efficiently implement searching for documents under a search criteria(such as presence or absence of a hidden combination of hidden keywords)under various cryptographic assumption.

Our results can be viewed in a variety of ways:as a generalization of notion of a Private Information Retrieval(to the more general queries and to a streaming environment as well as to public-key program);as positive results on privacy-preserving data mining and as a delegation of hidden program computation to other machines.

We introduced the cloud computing is a emerging computing paradigm in which resources of the computing infrastructure are provided as service over the internet.

As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud severs,which are not within the same trusted domain as data owners.To keep sensitive user data confidential against untrusted servers,existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users.However,in doing so,these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when finegrained data access control is desired and thus do not scale well.

The problem of simultaneously achieving finegrainedness, scalability and data confidentiality of access control actually still remains unsolved. This project address this challenging open issue by,on one hand,defining and enforcing access policies based on data attributes and,on the other hand, allowing data owner to delegate most of the computation tasks involved in finegrained data access control to untrusted cloud serverswithout disclosing the underlying data contents.

We achieve this goal by exploiting and uniquely combining techniques of attribute-based encryption ,lazy re-encryptionand proxy re-encryption.Our proposed scheme has salient properties of user access privilege confidentiality and user secret key accountability.

Extensive analysis that our proposed scheme is highly efficient and provable secure under existing security models.

3. SYSTEM ANALYSIS

In our Existing system private keyword-based file retrieval scheme that was originally proposed by Ostrovsky. Their scheme allows a user to retrieval files of interest from an untrusted server without leaking information. In Ostrovsky scheme, which relies on a public key cryptosystem, the Paillier cryptosystem.The main drawback of Ostrovsky scheme is that it will cause a heavy querying overhead incurred on the cloud, and thus goes against the original intention of cost efficiency.[2] However , the Ostrovsky scheme 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 interest to that user. It will quickly become a performance bottleneck when the cloud needs to process thousands of queries over a collection of hundreds of thousands of files.

In our proposed 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 basic idea of EIRQ is to construct a privacy preserving mask matrix that allows the cloud to filter 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 knowing anything about user privacy.[8] Focusing on different design goals, we provide two extensions: the first extension emphasizes 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.

The main advantage of EIRQ scheme is, 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.

4.SYSTEM DESIGN

This EIRQ scheme consists of three entities: the Aggregation and Distribution Layer(ADL), users, and the cloud.

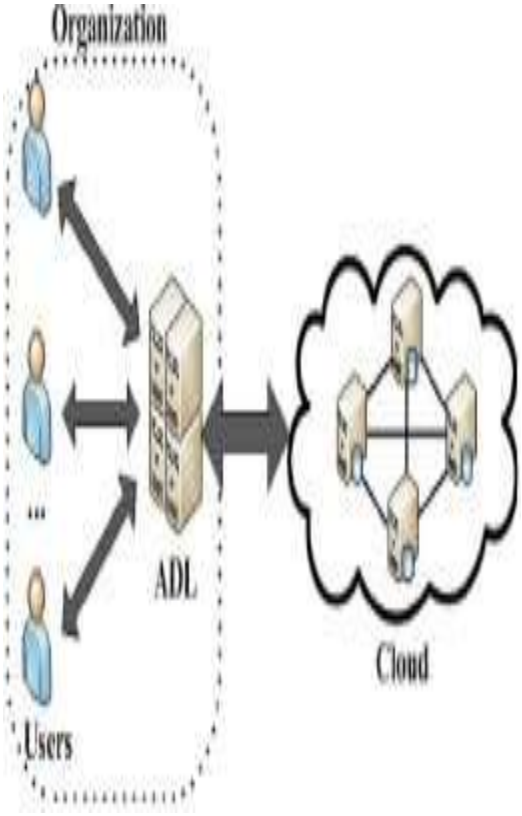

Fig: System Architecture

The user send their queries to the ADL, which will aggregate user queries and send a combined query to the cloud. Then the cloud processes the 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.[7]-[8]The Differential query service provided by allowing each user to retrieve matched files on demand. A user selects a particular rank for his query to determine the percentage of matched files to be returned.

5. MODULES IMPLEMENTATION

Differential Query services

We introduce a novel concept,*differential query services* to COPS,where the users are allowed to personally decide how many matched files will be returned[5]-[8]. This is motivated by the fact that under certain cases,there are lot of files matching a user's query,but the user is interested in only acertain percentage of matched files.Therefore,by

allowing the users to retrieve matched files on demand,the

Efficient Infromation Retrieval For Ranked Query
In this module,each use can choose the rank of his query to determine 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 filter 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 knowing anything about user privacy[2]-[8].Focusing on different design goals,we provide two extensions:the first extension emphasizes 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.

Aggregation And Distribution Layer
An ADL is deployed in an organization that authorizes its staff to share data in the cloud.The staff members,as the authorized users,send their queries to the ADL,which will aggregrate 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 distribute the search results to each user. To aggregrate 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.

Ranked Queries

6. SYSTEM PRELIMINARIES

We have three EIRQ schemes are used.

- EIRQ-Efficient Scheme
- EIRQ-Simple Scheme
- EIRQ-Privacy 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-1 queries have the highest rank and Rank-r queries have the lowest rank. We basically determine the relationship by allowing Rank-i queries to retrieve (1-i/r) percent of matched files[8]. Therefore Rank-1 queries can retrieve 100% of

else

M[*i,j*] = Epk(0) adjust and β so that file survival rate is 1

FileFilter (run by the cloud)for

bandwidth consumed in the cloud can be largely reduced.

To further reduce the communication cost, a differential query service is provided by allowing each user to retrieve matched files on demand. Specifically, a user selects a for particular *rank* his query to determine 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 needs a small subset of them. 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.[2]-[6]. After aggregating enough user queries the ADL runs the Matrix Construct algorithm to send a mask matrix to the cloud. keywords. If the file rank is i, then the probability of being filtered out is i/r[2]. Therefore, Rank-0 files will be mapped into a buffer with probability 1 and Rank-r files each keyword by the highest rank of queries choosing it and then rank each file by the highest ranks of its

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. queries matching this file. We simply determine the probability of a file being returned by the highest percentage of matched files. Specifically we first rank probability of a file being returned by the highest rank of returned and which will not.We simply resolve the

matched files and Rank-r queries cannot retrieve any files.
Secondly, we should conclude which files will be

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*** <= *r –l* **then** M[*i,j]* = *Epk*(1)

each file F*j* stored in the cloud **do**
for *i*=1 to *d* **do** F |
k = *j* mod *r* ; C*j* = Dic[*i*]_F*j* M[*i,k*];e*j*= C*jj*

map (c*j*,e*j*) times to a buffer of siz

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 project 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
are in each rank, nor which users are in which ranks.

Therefore, EIRQ-Simple can protect the basic level of rank privacy for a user. -Privacy working process is similar to Fig.2b. Intuitively, EIRQ-Privacy adopts one buffer, with different mapping times for files of different

7.EXPERIMENTAL EVALUATION

In this section, we will compare three EIRQ schemes from the following aspects:computation and communication cost incurred on the cloud. Then based on the simulation results,we deploy our program in Amazon Elastic Compute Cloud(EC2) to test the transfer-in and transfer-out time incurred on the cloud when executing private searches. Note that the energy-performance trade-off is crucial to the success of cloud computing and existing energy-saving techniques are hard to directly extend to a cloud environment[19].As part of our future extensions ,we will evaluate the consumed energy overhead in the cloud to verify the effectiveness of our schemes.

a)Computional cost In section, the computional cost is mainly determined by the number of exponentations performed by the cloud, which is almost the same under the Bloom filter and the Ostrovsky parameter settings[3]-[8]. In order to justify the analyses, we will compare the computational cost between No Rank and three EIRQ
b)Communication cost

MatrixConstruct and FileFilter.
In EIRQ-Simple working process is similar to Fig.2b. Intuitively, given queries that are classified into 0~r ranks, ADL sends r combined queries denoted as Qr…….Qr-1 to the cloud, each with a different rank[1]-[5].The messages from the ADL to the cloud are r encrypted
In EIRQ queries, the buffer size, and the mapping times, where r is the information. Given r,the cloud only knows the number of query ranks without knowing how many users

ranks[2][5].The message from the ADL to the cloud is a d-row and m-column mask matrix, where d is the number of keywords in the dictionary. Therefore, EIRQ-Privacy provides a high level of user rank privacy.

It depends on the buffer size generated by the cloud in different ways under different parameter settings.Futhermore; the buffersize depends on the number of flies that match queries, which is different when users have different common interests.

8.CONCLUSION

In this project,we proposed three EIRQ schemes based on an ADL to provide differential query services while 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 scheme

9. REFERENCE

[1] R.Ostrovsky and W.Skeith "Private Searching on Streaming Data," in Proc. CRYPTO, 2005, pp. 233-240.
[2] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," in Proc. ACM CCS, 2006, pp. 79-88.
[3] J. Bethencourt, D. Song, and B. Waters, "New Constructions and Practical Applications for Private Stream Searching," in Proc. IEEE SP, 2006, pp. 1-6.
[4] R. Ostrovsky and W. Skeith, "Private Searching on Streaming Data," J. Cryptol., vol.20, no.4, pp. 397-430, Oct. 2007.
[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, Fine-Grained Data Access Control in Cloud Computing," in Proc. IEEE INFOCOM, 2010, pp.
[6] P. Mell and T. Grance, "The NIST definition of Cloud Computing(Draft)," in NIST Special Publication. Gaithersburg, MD, USA:

National Institute of Standards and Technology, 2011.

[7] X. Yi and E. Bertino, "Private Searching for Single and Conjunctive Keywords on Streaming Data," in Proc. ACM Workshop Privacy Electron, Soc., 2011, pp. 153-158.

[8] Q. Liu, C.C. Tan, J. Wu, and G. Wang, "Efficient Information Retrieval for Ranked Queries in Cost-Effective Cloud Environments," in Proc. IEEE INFOCOM, 2012, pp.2581-2585.

.