ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

Cost proficient clouds for enquiry services

Santosh Chidambar Deshpande

M.Tech 4th Semester Computer Science, SDIT, Manglore, India

Abstract: Cloud computing as a developing technology drift is expected to restructure the advances in information technology. In a cost-proficient cloud location, a user can bear a certain degree of interruption while retrieving information from the cloud to reduce costs. In this paper, we address two important concerns in such an location: discretion and proficiency. We analyze a remote keyword based data retrieval system that was introduced by Ostrovsky. Their system allows a user to retrieve data of interest from an untrusted third party without leaking any information. The disadvantage is that it will cause a thick inquiring overhead acquired on the cloud and thus goes against the original purpose of cost proficiency. In this paper, we present three effective information retrieval for graded inquiry systems to decrease inquiring overhead incurred on the cloud. In, effective information retrieval for graded inquiry system inquiries are categorized into many grades, where a higher graded inquiry can retrieve a higher percentage of matched data. A user can retrieve data on demand by choosing inquiries of dissimilar grades. This feature is beneficial when there are a large number of matched data, but the user only needs a small subsection of them. Under different limitation settings, wide spread appraisals have been conducted on both analytical models and on a real cloud location, in order to examine the efficiency of our systems.

Keywords: cloud computing, budget adeptness, discrepancy interrogation amenities, discretion.

I. INTRODUCTION

Cloud computing as a developing technology is predictable to restructure information technology procedures [1]. Due to the irresistible merits of cloud computing, e.g., cost-efficiency, litheness and scalability, more and more administrations choose to outsource their data for membership in the cloud. As a typical cloud application, an administration subscribes the cloud amenities and authorizes its staff to share data in the cloud. Each files described by a set of keywords, and the staff, as authorized users, can retrieve data of their interests by inquiring the cloud with certain keywords. In such anlocation, how to protect user secrecy from the cloud, which is a third party outside the security boundary of the of the administration, becomes a key problem.

User secrecy can be classified into search secrecy and access secrecy [2]. Search secrecy means that the cloud knows nothing about what the user is searching for, and access secrecy means that the cloud knows nothing about which data are returned to the user. When the data are stored in the clear forms, a naive solution to protect user secrecy is for the user to request all of the data from the cloud; this way, the cloud cannot know which data the user is really interested in.

While this does provide the necessary secrecy, the communication cost is high. Remote searching was proposed by Ostrovsky et al. [3], [4] (referred to as the Ostrovsky methods in this paper), which allows a user to retrieve data of interest from an untrusted server without leaking any information. However, the Ostrovsky methods has a high computational cost, since it requires the cloud to process the inquiry (perform homomorphism encryption) on every file in a collection. Otherwise, the cloud will learn that certain data, without processing, are of no interest to the 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 data. We argue that subsequently proposed improvements, like [5], [6], also have the same drawback. Commercial clouds follow a pay-as-you-go model, where the customer is billed for different operations such as bandwidth, CPU time, and so on. Solutions that incur excessive computation and communication costs are unacceptable to customers.

To make remote searching applicable in a cloud location, our previous work [7] designed a cooperate remote searching protocol (COPS), where a proxy server, called the combination and scattering layer (CSL), is introduced between the users and the cloud. The CSL deployed inside an administration has two main functionalities: aggregating user queries and distributing search results. Under the CSL, the computation cost incurred on the cloud can be largely reduced, since the cloud only needs to execute a combined inquiry once, no matter how many users are executing queries. Furthermore, the communication cost incurred on the cloud will also be reduced, since data shared by the users need to be returned only once. Most importantly, by using a series of secure functions, COPS can protect user secrecy from the CSL, the cloud, and other users.

In this paper, we introduce a novel concept, differential inquiry amenities, to COPS, where the users are allowed to personally decide how many matched data will be returned. This is motivated by the fact that under certain cases, there are a lot of data matching a user's inquiry, but the user is Interested in only a certain percentage of

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

matched data. To illustrate, let us assume that Alice wants to retrieve 2 percent of the data that contain keywords ''A, B'', and Bob wants to retrieve 20 percent of the data that contain keywords ''A, C''. The cloud holds 1,000 data, where fF1; . . . ; F500g and fF501; . . . ; F1000g are described by keywords ''A, B'' and ''A, C'', respectively. In the Ostrovsky methods, the cloud will have to return 2,000 data. In the COPS methods, the cloud will have to return 1,000 data. In our methods, the cloud only needs to return 200 data. Therefore, by allowing the users to retrieve matched data on demand, the bandwidth consumed in the cloud can be largely reduced. Motivated by this goal, we propose a method, effective information retrieval for graded inquiry systems (EIRGIS), in which each user can choose the rank of his inquiry to determine the percentage of matched data to be returned. The basic idea of EIRGIS is to construct a secrecy-preserving mask matrix that allows the cloud to filter out a certain percentage of matched data before returning to the CSL. This is not a trivial work, since the cloud needs to correctly filter out data according to the rank of queries without knowing anything about user secrecy. Focusing on different design goals, we provide two extensions: the first extension emphasizes simplicity by requiring the least amount of modifications from the Ostrovsky methods, and the second extension emphasizes secrecy by leaking the least amount of information to the cloud.

Our key contributions are as follows:

1. We propose three EIRGIS methods based on the CSL to provide a cost-efficient solution for remote searching in cloud computing.
2. The EIRGIS methods can protect user secrecy while providing a differential inquiry service that allows each user to retrieve matched data on demand.
3. We provide two solutions to adjust related parameters; one is based on the Ostrovsky methods, and the other is based on Bloom filters.
4. Extensive experiments were performed using a combination of simulations and real cloud deployments to validate our methods.

The remainder of this paper is organized as follows. We introduce related work in Section 2 before presenting preliminaries in Section 3. We describe EIRGIS methods in Section 4 and adjust the parameters in Section 5. After analyzing the performance and security of the proposed methods in Section 6, we conduct evaluations in Section 7. Finally, we conclude this paper in Section 8.

II. RELATED WORKS

Our work aims to provide differential inquiry amenities while protecting user secrecy from the cloud. Existing research that is similar to ours can be found in the areas of remote searching [3], [4], [5], [6], [7], [8], [9], [10], [11]. Unlike searchable encryption [2], [12], where the user conducts searches on encrypted data, remote searching performs keyword-based searches on unencrypted data. Remote searching was first proposed in [3], [4], which allows a server to filter streaming data without compromising user secrecy. Their solution requires the server to return a buffer of size Oðf logðfÞÞ when f data match a user's inquiry. Each file is associated with a survival rate, which denotes the probability of this file being successfully recovered by the user. Based on the Paillier cryptosystem [13], the data that mismatch a inquiry will not survive in the buffer, but the matched data enjoy a high survival rate.

Among various extensions, [5], [6] further reduced the communication cost from O(f log(f)) to O(f) by solving a set of linear equations to recover f matched data. However, their methods requires the decryption of one more buffer, thus the computation cost is higher than the Ostrovsky methods. Reference [8] presented an efficient decoding mechanism which allows the recovery of data that collide in a buffer position. Reference [9] proposed a recursive extraction mechanism, which requires a buffer of size O (f) when f data match a user's inquiry. Reference [10] proposed two new communication-optimal constructions; one uses Reed-Solomon codes and allows for a zero-error, and the other is based on irregular LDPC codes and allows for lower computation cost at the server. The above remote searching methods only support searching for OR of keywords or AND of two sets of keywords. Reference [11]extended the types of queries to support disjunctive normal forms (DNF) of keywords. The main drawback of existing remote searching methods is that both the computation and communication costs grow linearly with the number of users executing queries. Thus, when applying these methods to a large-scale cloud location, inquiring costs will be extensive.

Our previous work [7] was the first to make remote searching techniques applicable to a cloud location. However, [7] requires the cloud to return all of the matched data, which may cause a waste of bandwidth when only a small percentage of data are of interest. To alleviate the problem, we introduced the concept of differential inquiry amenities in [14]. The main difference between this work and[14] is that we provide two extensions to address different aspects of the problem, and we conduct extensive experimentson a real cloud to verify the effectiveness of the proposed methods.

III. BACKGROUNDS

1. System Model

The system mainly consists of three entities:1 the combination and scattering layer (CSL), many users, and the cloud, as shown in Fig. 1. For ease of explanation, we only use a single CSL in this paper, but multiple CSLs can be deployed as necessary. ACSL is deployed in an administration that authorizes its staff to share data in the cloud. The staff members, as the authorized users, send their queries to the CSL, which will aggregate user queries and send a combined

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

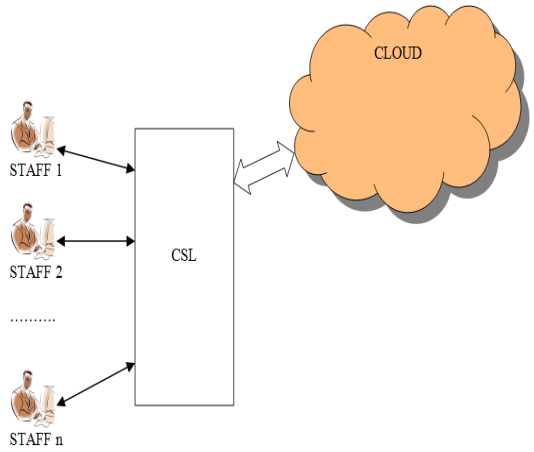


Fig 1

inquiry to the cloud. Then, the cloud processes the combined inquiry on the file collection and returns a buffer that contains all of matched data to the CSL, which will distribute the search results to each user. To aggregate sufficient queries, the administration may require the CSL to wait for a period

2. *Security Model and Design Goals*

The CSL is deployed inside the security boundary of an administration, and thus it is assumed to be trusted by all of the users. In the supplementary file available online, we will discuss how the EIRGIS methods work without such an assumption. The communication channels are assumed to be secured under existing security protocols, such as SSL, during information transfer. With these assumptions, as long as the CSL obeys our methods, a user cannot know anything about other users' interests, and thus the cloud is the only attacker in our security model. As in existing work [15], [16], the cloud is assumed to be honest but curious. That is, it will obey our methods, but still wants to know some additional information about user secrecy.

Reference [2] classified user secrecy into search secrecy and access secrecy. In our work, user queries are classified into multiple ranks, and thus a new kind of user secrecy, rank secrecy, also needs to be protected against the cloud. Rank secrecy entails hiding the rank of each user inquiry from the cloud, i.e., the cloud provides differential inquiry amenities without knowing which level of service is chosen by the user. Rank secrecy can be classified into basic level and high level, where basic level will hide the rank of each inquiry from the cloud, and the high level will further hide the number of ranks from the cloud. Our design goal can be subdivided as follows:

- Cost efficiency. The users can retrieve matched data on demand to further reduce the communication costs incurred on the cloud.
- User secrecy. The cloud cannot know anything about the user's search secrecy, access secrecy, and at least the basic level of rank secrecy.

3. *Overview of the Ostrovsky Methods*

We briefly introduce the Ostrovsky methods [3], [4], which relies on a public key cryptosystem, the Paillier cryptosystem [13]. Let Epk (m) denote the encryption of plaintext m under public key pk. The Paillier cryptosystem has the following homomorphism properties:

- Epk(a).Epk(b)=Epk(a+b)
- Epk(a)b=Epk(a.b).

The Paillier cryptosystem allows the performance of certain operations, such as multiplication and exponentiation, on cipher text directly. Given the resultant cipher text, the user can obtain the corresponding plaintext that processes addition and multiplication operations.

The Ostrovsky methods consist of three algorithms, the working process of which is shown in Fig. 2a. Two assumptions are used in their methods: first, a dictionary that consists of the universal keywords is assumed to be publicly available; second, the users are assumed to have the ability to estimate the number of data that match their queries. To better illustrate its working process, we provide an example in the supplementary file available online.

Step 1. The user runs the Generate Inquiry algorithm to send an encrypted inquiry to the cloud. The inquiry is a bit string encrypted under the user's public key, where each bit is an encryption of1, if the keyword in the dictionary is chosen; otherwise, it is an encryption of 0.

Step 2. The cloud runs the Remote Search algorithm to return an encrypted buffer to the user. Generally speaking, the cloud processes the encrypted inquiry on every file in the collection to generate an encrypted c-e pair, and maps it to multiple entries of an encrypted buffer. For file Fj, the corresponding c-e pair, denoted as (cj; ej), is generated as follows: the bits in inquiry Q corresponding to keywords in Fj are multiplied together to

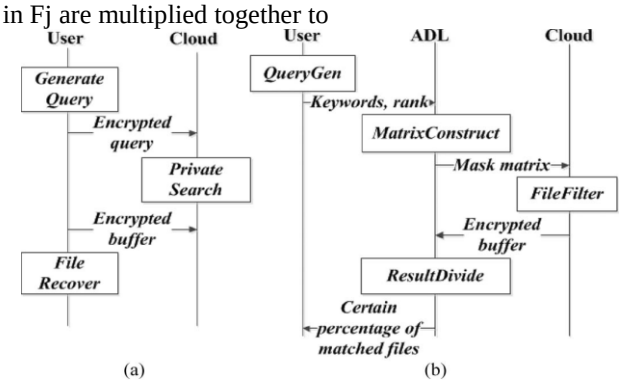


Fig.2. Working process. (a) Ostrovsky methods. (b) EIRGIS-Efficient methods.

Step 3. The user runs the File Recover algorithm to recover data. The user decrypts the buffer, entry by entry, to obtain the plaintext c-e pairs. For the entries in the

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

survival state, file content can be recovered by dividing the plaintext e value by the plaintext c value.

The security of the Ostrovsky methods derives from the semantic security of the Paillier cryptosystem. The key technique of their methods is that the data mismatching a user's inquiry are processed to encrypted 0s, which have no impact on the matched data, even if they are mapped in the same entry. Thus, the buffer size only depends on the number of matched data, which is much smaller than the number of data stored in the cloud.

IV. METHODS DESCRIPTION

In this section, we will describe the original EIQR methods and its two extensions. To distinguish the three EIRGIS methods, we name the original EIRGIS methods as EIRGIS Efficient the first extension as EIRGIS-Simple, and the second extension as EIRGIS-Secrecy, in this paper.

The basic idea of EIQR-Efficient is to construct a secrecy-preserving mask matrix with which the cloud can filter out a certain percentage of matched data before mapping them to a buffer. As proven in the Ostrovsky methods, the file survival rate is determined by the buffer size and mapping times. Therefore, the basic idea of two extensions is that, for each rank i 2 f0; . . . ; rg, the CSL adjusts the buffer size i and the mapping times i to make the file survival rate qi approach 1 i=r. To better illustrate the working process of the EIRGIS methods, we provide examples in the supplementary file available online.

A. The EIRGIS-Efficient Methods

Before illustrating EIQR-Efficient, two fundamental problems should be resolved:

Firstly, we should determine the relationship between inquiry rank and the percentage of matched data to be 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 simply determine this relationship by allowing Rank-i queries to retrieve ð1 i=rÞ percent of matched data. Therefore, Rank-0 queries can retrieve 100 percent of matched data, and Rank-r queries cannot retrieve any data.

Secondly, we should determine which matched data 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. Specifically, 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. If the file rank is i, then the probability of being filtered out is i=r. Therefore, Rank-0 data will be mapped into a buffer with probability 1, and Rank-r data will not be mapped at all. Since unneeded data have been filtered out before mapping, the mapped data should survive in the buffer with probability 1. In Section 5, we will illustrate how to adjust the buffer size and mapping times to achieve this goal.

EIRGIS-Efficient mainly consists of four algorithms, with its working process being shown in Fig. 2b. Since algorithms Inquiries and Result Divide are easily understood, we only provide the details of algorithms Matrix- Construct and File Filter in Alg. 1.

Step 1. The user runs the Inquiry Gen algorithm to send keywords and the rank of the inquiry to the CSL. Since the CSL is assumed to be a trusted third party, this inquiry will be sent without encryption.

Step 2. After aggregating enough user queries, the CSL runs the Matrix Construct algorithm to send a mask matrix to the cloud. The mask matrix M is a d-row and r-column matrix, where d is the number of keywords in the dictionary, and r is the lowest inquiry rank. Let M½i; j denote the element in the

Scheme	Computation	Communication under Ostrovsky setting	Communication under Bloom filter setting [2]
No Rank	$O(t)$	$O(d+f\cdot\log(f/\alpha))$	$O(d+f\cdot\log_{0.6185}(\alpha))$
EIRQ-Simple	$O(r\cdot t)$	$O(r\cdot d+\sum_{i=0}^{r}(f_i\cdot\log(f_i/(\alpha+i/r))))$	$O(\sum_{i=0}^{r}(r\cdot d+f_i\cdot\log_{0.6185}(i/r+\alpha)))$
EIRQ-Privacy	$O(\max(\gamma_i)\cdot t)$	$O(\max(\gamma_i)\cdot d+\sum_{i=0}^{r}(f_i'\cdot\log(f_i'/(\alpha+i/r))))$	$O(\max(\gamma_i)\cdot d+\sum_{i=0}^{r}(f_i'\cdot\log_{0.6185}(i/r+\alpha)))$
EIRQ-Efficient	$O(t)$	$O(r\cdot d+\sum_{i=0}^{r}(f_i'\cdot(1-i/r))\cdot\log(\frac{\sum_{i=0}^{r}f_i'\cdot(1-i/r)}{\alpha}))$	$O(r\cdot d+\sum_{i=0}^{r}(f_i'\cdot(1-i/r))\cdot\log_{0.6185}(\alpha))$

i-th row and the j-th column, and let l be the highest rank of queries that choose the i-th keyword Dic½I in the dictionary. M is constructed as follows: for the i-th row of M that corresponds to set to 0, then each element is encrypted under the CSL's public key pk. For the rows that correspond to Rank-l keywords, the CSL sets the first r _ l elements, rather than random r _ l elements, to 1. There as on is to ensure that, given any Rank-l file Fj, when we choose a random number k, the probability of all of the k-th elements of the rows that correspond Fj's keywords being0 is l=r, which is determined by the highest rank of Fj's keywords.

Step 3. The cloud runs the File Filter algorithm to return a buffer that contains a certain percentage of matched data to the CSL. Specifically, the cloud multiplies the k-th elements of the rows that correspond to Fj's keywords together ro form cj, where k ¼ j mod r. Then, it powers jFjj to cj to obtain ej, and maps the c-repair into multiple entries of a buffer, as in the Ostrovsky methods. Note that, with Step 2, we can make sure that, for a Rank-l file Fj, the probability of cj being 0 is l=r, and thus the probability of Fj being filtered out is l=r

Step 4. The CSL runs the Result Divide algorithm to distribute search results to each user. File contents are recovered as the file Recover algorithm in the Ostrovsky methods. To allow the CSL to distribute data correctly, were quire the cloud to attach keywords to the file content. Thus, the CSL can find out all of the data that match users' queries by executing keyword searches.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

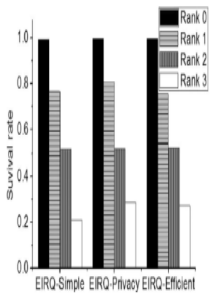
B. The EIRGIS-Simple Methods

The working process of EIRGIS-Simple is similar to Fig. 2b.Themain differences lie in the Matrix Construct and File Filter algorithms (see Alg. 2). Intuitively, given queries that are classified into 0 _ r ranks, CSL sends r combined queries, denoted as Q0; . . .;Qr_1, to the cloud, each with a different rank. Specifically, for Qi, the CSL sets the j-th bit to an encryption of 1 if the j-th keyword Dic½j_ in the dictionary is chosen by at least one Rank-i inquiry. The cloud then will generate r buffers, denoted as B0; . . . ;Br_1, each with a different file survival rate. Specifically, for Bi, the CSL adjusts the mapping time _i and the buffer size _i so that the survival rate of data in Bi is qi ¼ 1 _ i=r, where0 _ i _ r _ 1.

The main drawback of EIRGIS-simple is that it returns redundant data when there is data satisfying more than one ranked inquiry. For example, if Fi is of interest by Rank-0and Rank-1 queries, it will be returned twice (in Rank-0buffer and Rank-1 buffer, respectively), which wastes the network bandwidth. Therefore, the best case scenario is when there are no data of interest to different rank equerries, and the worst case scenario is when queries of different ranks inquiry the same data.

TABLE 2
Parameters

Notation	Description	Value
$\|F\|$	File content	$1KB$
$\|w\|$	Keyword content	$1KB$
n	The number of users	1-100
d	The number of keywords in *Dic*	100
k	The number of keywords in each query	1-5
w	The number of keywords in each file	1-5
t	The number of files stored in the cloud	1, 000
r	The lowest user rank	4
α	Threshold value	0.1



C. The EIRGIS-Secrecy Methods

The working process of EIRGIS-Secrecy is similar to Fig. 2b.The main differences lie in the Matrix Construct and File Filter algorithms (see Alg. 3). Intuitively, EIRGIS-Secrecy adopts one buffer, with different mapping times for data of different ranks. Let _i denote the mapping times for a Rank-I inquiry, and let l be the highest rank of queries that choose the i-th keyword Dic½i_ in the dictionary. The mask matrix M is a d-row and m-column matrix, where d is the number of keywords in the dictionary, and m ¼ max _i. The Matrix Construct algorithm constructs M in the following way: for the i-th row of M that corresponds to, the CSL sets and to 0, and then encrypts each element under its public key. Note that for a row that corresponds to a Rank-l keyword, the CSL sets the first_l elements, rather than random _l elements, to 1. The reason is to ensure that, given any Rank-l file, when we multiply the rows that correspond to file keywords together in a element-by-element way, the resulting row contains _l elements whose values are larger than 0.

In the File Filter algorithm, for each file Fj, the cloud multiplies the rows that correspond to file keywords, element by element, to form a resulting row. Each element in the resulting row corresponds to a c value. Letcj;1; . . . ; cj;m denote Fj's c values, where m ¼ max _i. The cloud powers the file content jFjj to cj;k to form ej;k, and maps ðcj;k; ej;kÞ to the buffer once, where 1 _ k _ m. Note that with the Matrix Construct algorithm, we can make sure that, for a Rank-l file, the number of c values larger than 0 is Therefore, although m c-e pairs will be mapped, only of them will take effect, which is equal to mapping c-e pairs times to a buffer.

V. ANALYSES

A. Security Analysis

We will show that EIRGIS methods can provide search secrecy, access secrecy, and rank secrecy as follows.

1. Search Secrecy:

In the three methods, the combined inquiry sent to the cloud is encrypted under the CSL's public key with the Paillier cryptosystem. The inquiry is a matrix of encrypted 0s and 1s. The Paillier cryptosystem is semantically secure, and the cipher text of every 1 or 0 is different from other 1s or 0s. Therefore, the cloud cannot deduce what each user is searching for from the encrypted inquiry.

2. Access Secrecy:

In the three methods, the cloud processes the encrypted inquiry on each file in a collection, and maps the processing result into a buffer, which is encrypted with the CSL's public key. The cloud conducts this process for all data in the same way. Therefore, the cloud cannot know which data are actually returned from the encrypted buffer.

3. Rank Secrecy:

In EIRGIS-Simple, the messages from the CSL to the cloud are r encrypted queries, the buffer size, and the mapping times, where r is the information, which we leak more than [3]. Given r, the cloud only knows the number of inquiry ranks without knowing how many users are in each rank, nor which users are in which ranks.

Therefore, EIRGIS Simple can protect the basic level of rank secrecy for a user. In EIRGIS-Secrecy, the message from the CSL to the cloud is a d-row and m-column mask matrix, where d is the number of keywords in the dictionary, and m ¼ max i is the maximal value of mapping times.

Here, no extra information is leaked more than [3]. Therefore, EIRGIS-Secrecy provides a high level of user rank secrecy. In EIRGIS-Efficient, the message from the CSL to the cloud is a d-row and r column mask matrix, where d is the number of keywords in the dictionary, and r is the lowest rank of user queries.

Here, r is the information that we leak more than [3]. Therefore, EIRGIS-Efficient can protect the basic level of rank secrecy

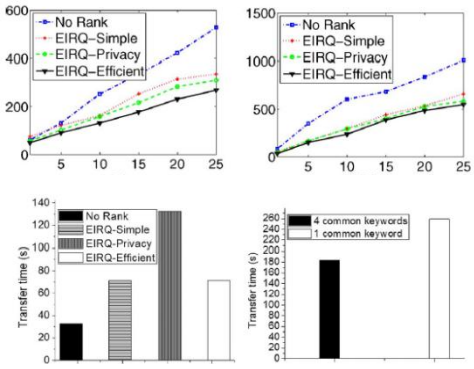
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

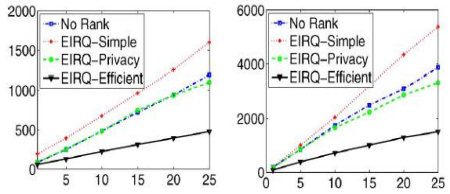
National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015



5.2 Performance Analysis



We compare the performance between No Rank and the three EIRGIS methods under different parameter settings (see Table 1). In No Rank, the CSL only combines user queries, but does not provide differential inquiry amenities. In the supplementary file available online, we also provide a comparison of performance between No Rank and the work in [3], [6]. Suppose that queries are classified into 0 r ranks, t data stored in the cloud whose keywords constitute a dictionary of size d, fi data matching Rank-I queries, and f0 i data matching Rank-i queries but mismatching higher ranked queries. Furthermore, in No Rank and EIRGIS-Efficient, the threshold file survival rate p0 is set to; in EIRGIS-Simple and EIRGIS-Secrecy, p0 i is set to i=r þ

B. Computational Cost

We only consider the cost of the exponential operation, which is the most expensive. In both parameter settings, the results are the same. In EIRGIS-Simple, the computational cost is r times more than No Rank since, for each ranked inquiry, the cloud needs to process it on the file collection once. In EIRGIS-Secrecy, the computational cost is maxiÞ times more than No Rank since, for each file, the cloud needs to execute maxiÞ exponentiations with the matrix elements. In EIRGIS-Efficient, the computational cost is much the same as in No Rank, since the cloud needs to execute exponentiation once for each file.

VI. CONCLUSION

In this paper, we proposed three EIRGIS methods based on an CSL to provide differential inquiry amenities while protecting user secrecy. By using our methods, a user can retrieve different percentages of matched data by specifying queries of different ranks. By further reducing the communication cost incurred on the cloud, the EIRGIS methods make the remote searching technique more applicable to a cost-efficient cloud envir onment. However, in the EIRGIS methods, 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 EIRGIS methods.

REFERENCES

[1] 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.
[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] R. Ostrovsky and W. Skeith, ''Remote Searching on Streaming Data,'' in Proc. CRYPTO, 2005, pp. 233-240.
[4] R. Ostrovsky and W. Skeith, ''Remote Searching on Streaming Data,'' J. Cryptol., vol. 20, no. 4, pp. 397-430, Oct. 2007.
[5] J. Bethencourt, D. Song, and B. Waters, ''New Constructions and Practical Applications for Remote Stream Searching,'' in Proc. IEEE SP, 2006, pp. 1-6.
[6] J. Bethencourt, D. Song, and B. Waters, ''New Techniques for Remote Stream Searching,'' ACM Trans. Inf. Syst. Security, vol. 12, no. 3, p. 16, Jan. 2009.
[7] Q. Liu, C. Tan, J. Wu, and G. Wang, ''Cooperative Remote Searching in Clouds,'' J. Parallel Distrib.Comput., vol. 72, no. 8, pp. 1019-1031, Aug. 2012.
[8] G. Danezis and C. Diaz, ''Improving the Decoding Efficiency of Remote Search,'' Int'l Assoc. Cryptol. Res., IACR Eprint Archive No. 024, SchlossDagstuhl, Germany, 2006.
[9] G. Danezis and C. Diaz, ''Space-Efficient Remote Search with Applications to Rateless Codes,'' in Proc. Financial Cryptogr. Data Security, 2007, pp. 148-162.
[10] M. Finiasz and K. Ramchandran, ''Remote Stream Search at the Same Communication Cost as a Regular Search: Role of LDPC Codes,'' in Proc. IEEE ISIT, 2012, pp. 2556-2560.
[11] X. Yi and E. Bertino, ''Remote Searching for Single and Conjunctive Keywords on Streaming Data,'' in Proc. ACM WorkshopSecrecy Electron. Soc., 2011, pp. 153-158.
[12] B. Hore, E.-C. Chang, M.H. Diallo, and S. Mehrotra, ''Indexing Encrypted Documents for Supporting Efficient Keyword Search,'' in Proc. Secure Data Manage., 2012, pp. 93-110.
[13] P. Paillier, ''Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,'' in Proc. EUROCRYPT, 1999, pp. 223-238.
[14] Q. Liu, C.C. Tan, J. Wu, and G. Wang, ''Efficient Information Retrieval for Ranked Queries in Cost-Effective Cloud Locations,'' in Proc. IEEE INFOCOM, 2012, pp. 2581-2585.
[15] S.Yu,C. Wang,K.Ren, andW. Lou, ''Achieving Secure, Scalable, Fine-Grained Data Access Control in Cloud Computing,'' in Proc. IEEE INFOCOM, 2010, pp. 1-9.
[16] G. Wang, Q. Liu, J. Wu, and M. Guo, ''Hierarchical Attribute-Based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers,'' Comput. Security, vol. 30, no. 5, pp. 320-331, July 2011.
[17] M. Mitzenmacher, ''Compressed Bloom Filters,'' IEEE/ACM Trans. Netw., vol. 10, no. 5, pp. 604-612, Oct. 2002.
[18] D. Guo, J. Wu, H. Chen, and X. Luo, ''Theory and Network Applications of Dynamic Bloom Filters,'' in Proc. IEEE INFOCOM, 2006, pp. 1-12.

BIOGRAPHY

Mr.Santhosh C Deshpande is currenty studying M.Tech in Shree Devi Institute of Technology under VTU in Computer Science branch. He has completed Engineering in SKSVMACET, Laxmeshwar under VTU.