International Journal of Computer Networks and Communications Security

VOL. 3, NO. 7, JULY 2015, 291–297
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Attribute Based Secure Query Processing in Cloud with Privacy Homomorphism

Ms. RUPALI S.KHACHANE[1] and Dr. PRADEEP K.DESHMUKH[2]

[1, 2] Dept. of Computer Engineering, RajarshiShahu College of Engineering, Tathawde, Pune, India
Affiliated To SavitribaiPhule Pune University

E-mail: [1]rupali.khachane@gmail.com, [2]pkdeshmukh9@gmail.com

ABSTRACT

Different types of business organizations are benefitted due to convenient as well as secure working of Cloud Computing and Data Outsourcing. A cloud, data owner and client are intrinsic part of the system. Hence, secure query of users with privacy of data owners received a worldwide importance in modern days cloud computing and data management. A research had been carried out by many people in cloud computing and its security to preserve query processing data, privacy of data owners and its clients. A Privacy Homomorphism (PH) Technique is being used in the system to provide prominent security features to client. Here, PH emphasise to resolve the security of query processing from client side, cloud, with the k-NN on R-tree index query and distance re-coding algorithm. PH technique support to leverage performance parameter in cloud computing.

Keywords: *Privacy Homomorphism, Encrypted data, Decryption Data, K-Nearest Neighbour, Cipher Text, Plaintext, Cloud Security.*

1 INTRODUCTION

In cloud computing, data owner use data and querying services for outsourcing on the cloud data. During this process, data is the separate and private asset of the data owner, hence that must be protected against cloud and querying client. Query which is fired by the client may disclose the sensitive details information of the client. Hence should be protected in cloud and from data owners.

Therefore, one of the major problem in cloud computing is to protect both, data privacy and query privacy amongst the data owner, client, and cloud –refer Figure 1.

Social networking is one of the aspiring sectors facing such type of privacy problem [2]. Cloud Computing is new platform to deploy, manage and provide solution to various types of storage-platform-problems, using internet-based infrastructure. The services such as Goggle Docs, Amazon EC2,Microsoft Azure, and Online file storage etc. are the examples of cloud computing which used widely by number of users worldwide..

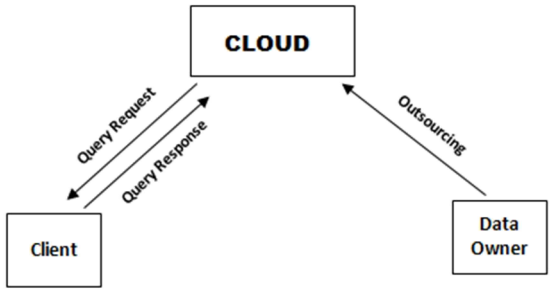

Fig. 1. General Model for query processing in Cloud

It is really sensitive issue to upload our personal data on the cloud because data privacy is a major concern and faces major security problem.

Sensitive information has to be encrypted before outsourcing, hence this creates an effective data utilization services, ultimately a big challenging task. One of the techniques of retrieval called Symmetric Searchable Encryption (SSE) of encrypted data on the cloud but still there is leakage of data privacy. Secure server side ranking, which is based on the order-preserving Encryption (OPE),

also includes the similarity relevance and robustness [3].

The rest of the paper is organized as follows. Section II reviews existing work on privacy-preserving query processing on outsourced data. Section III formulates the problem and section VI describe the challenges, Section V overviews the secure processing framework, and introduces ASM-PH, the privacy homomorphism used in this paper followed by detailed discussions on the protocols with a focus on distance-based queries. Section IX is explained the algorithms.

2 RELATED WORK

We will review existing data privacy-preserving outsourcing techniques for the purpose of query processing. In a common model, non-trusted outsourcing servers, stores and manages data on behalf of the data owners, who afterwards invites trusted users to put their query data. The first category of techniques is based on the generalization principle to minimize the disclosure of precise information.

For the privacy of the user and client data, various types of general solutions in recently research papers are deposited to showcase study on the data privacy. In recently done research papers, the most acceptable general solution is 'encryption'. Encryption means data deposited with service provider must be encrypted to avoid information leakage on the cloud. Agrawal et al [4], it is proposed one of the solutions so as to order preserving encryption scheme (OPES) by which, indexes can be built directly on cipher text. The various SQL statements such as MAX, MIN, COUNT, GROUP BY and ORDER BY can then be re-written and processed over the encrypted data. But OPES does not support SUM or AVG statements. In case of SUM and AVG, original data must be decrypted first. In private Information retrieval (PIR) for hiding a user query completely and to provide strong privacy and confidentiality, a query anonymisation usually used ask-Anonymity [5] and its variants to mix the user's query with other noisy query data.

In [6], [7], user privacy and data privacy is considered together. Yonghong Yu and WenyangBai discussed how to enforce data privacy and user privacy over outsourced database service. In [8], [9] proposed one of the solution based on secure traversal framework and privacy homomorphism based encryption scheme and secure protocols for processing k-nearest-neighbour queries (kNN) on R-tree index is given. In the authors following work [7], an integrated indexing technique with secure multiparty computation (SMC) based protocols to construct a secure index traversal framework is proposed and used.

To solve private processing of more specific queries, different techniques have been implemented, e.g. public data column and private data column are implemented by hashing in. But join by hashing is unable to retrieve other specific as well as relevant data columns. Some time before a paper published by researcher, proposes kNN queries by processing private & remotely using homomorphism encryption [2]. Theoretical protocols using homomorphic encryption have been proposed to process private document search by specific keywords in a line of documents .These protocols are still too costly to use practically. They perform only approximated search. Finally, we are not concerned to private query processing on outsourced encrypted data; although our data bucketization is inspired by the data bucketization idea in a work from that area [9].Our approach may also apply to protect query privacy in outsourced scenarios.

3 PROBLEM FORMULATION

In a cloud computing model, three parties' forms most important part, i.e. a data owner, querying client, and the cloud service provider (or simply the cloud). A data owner owns a huge data set D, and outsources its query processing service to the cloud.

The data set contains some proprietary and sensitive attributes θ like a salary, date of birth, social security number which needs to be protected from the cloud and the querying clients. While on the other hand, the client fires queries on the same sensitive attributes θ to retrieve the identifiers of qualified objects in D (data set). After the query processing, these identifiers can be used to retrieve non-sensitive contents like name, sexuality of these objects. The query q needs to be protected against both the data owner and the cloud. Hence, the summarized statement may be given here is that problem is to process queries on sensitive attributes.

4 CHALLENGES

In conventional query processing, current framework of the system has limitation such as computational limitation and communication limitation. In case of computational limitations, in each node traversal, there is a local distance computation on the client side and a decryption and recoding on the server side. As far as communication limitations are concerned, in each node traversal, both sides send and receive a set of

distances for the node entries. There are also several challenges regarding security and efficiency in this framework which are also clarified in next sessions.

1) The core of this framework is distance access which comprises local distance computation, decryption and recoding, and client scrambling.

2) Since each node traversal and distance access incurs both computational and communication limitations, optimization techniques will be designed to prune unnecessary distance computation and node traversal.

3) In case of security of data leakage, this framework preserves both data and query privacy, based on the security of ASM-PH. Nonetheless, it admitted certain amount of privacy loss in this framework, such as the disclosure of index topology to the client.

5 OVERVIEW OF PRIVACY HOMOMORPHISM: DESIGN

In this section, processing distance-based over a multidimensional can be treated as traversal on the tree nodes. It can be separated into two alternate procedures: Node traversal and Distance access. In the distance access which determines the next node to traverse based on the distances. It is computed form the current node and query point. To preserve client query and cloud data privacy, both procedures must remain secure in the outsourcing model of three parties. i.e., during the query processing neither data owner nor the cloud can identify the traversed nodes or obtain any type of information that can pinpoint the query point (such as the exact distances to the query point). In that time the client should not have an access to the actual node contents during distance access and the node traversal. Some of the algorithms to implement above scenario are given below:

1) Privacy-Preserving Processing Framework for Distance-based Queries

2) Recode: Distance Recoding Scheme

Scenario of search and retrieval over encrypted data, Consider a data management system hosting data service, as illustrated in Fig-2, in which three different entities are involved: data owner, data user and a storage server.

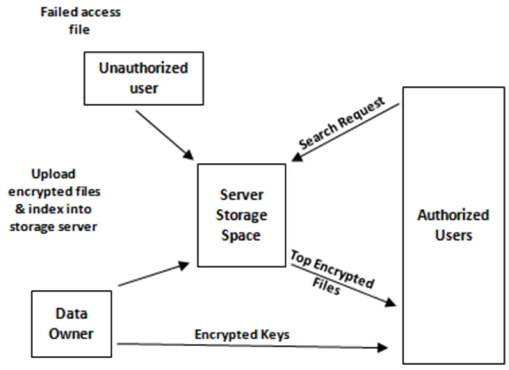

Fig. 2. Scenario of search and retrieval over encrypted data

The data owner has a collection of data files. Data owners are encouraged to outsource their data from local systems to global space for great flexibility. For protecting data files, they are encrypted before uploading into such global space. Thus enabling search and retrieval over such encrypted data is of paramount importance. The data owner has a collection of n files say, C = {f1, f2 ,...fn } which may be of extension .txt, .doc and .pdf. For protecting the file from the unauthorized person we need to apply different types of privacy homomorphism algorithms [7].

5.1 Secure Privacy Homomorphism (PH)

PH is an encryption transformation which maps a set of operations on plain text to another set of operations on cipher text.

$$E(A) - E(B) = E(I)$$
$$E(I) = E^{-1}(I)$$

1. Encryption

Converting plain text into a cipher text with using public and private keys. Consider Z is a set of plain text with using secret keys converted into cipher text. In fig-2 as per the paper, Client sends query requirement to Cloud then owner sends encrypted key index to Client.

Z=queries
E (I) = encrypted index key

2. Decryption

Converting cipher text into a plain text with using public and private keys. In fig-2, data owner sends the decryption index key E^{-1} (I) to the data cloud for future distance decryption.
E^{-1} (I) = decrypted index key.

6 SYSETM ARCHITECTURE

6.1 Privacy-Preserving Query Processing Framework

When processing distance-based queries, a multidimensional index can be treated as traversal on the tree nodes. Very clearly, this may be divided into two alternate processes i.e. node traversal and distance access.

The distance access determines the next node to traverse which is depending upon the distances computed from the current node and query point. To safeguard query and data privacy, both procedures must remain secure in the outsourcing model of three parties i.e. when query is being processing not only data owner but the cloud can identify the traversed nodes also or may obtain any information that may point out the query point as the exact distances to the query point. Till time, the client should have no access to the actual node contents during distance access and node traversal. Here, in fig-3, showing the framework of secure query processing. Whereas, other part is to protect data privacy, the client has only access to an encrypted version of the index, and must go ahead to process their query together with the cloud, which will decrypt the distances it, computes locally. The distance access is a collective procedure of the client and data cloud, in which not a single party has access to the actual distances [2].

The detailed process flow of this framework is as follows.

1. Sending query requests to cloud by client
   Data owner only allowed the authenticated user.
2. During this process data owner sends an encrypted variant of index E (I) to client , and shadow index E^{-1} (I) to cloud. In each index node, the key entry e.g. e_1, e_2, e_3 is encrypted by encryption scheme E(·),
3. Although the pointers e.g., p_1, p_2, p_3 are not encrypted. It means that, the index has common topology as the basic index but each key value is encrypted. The index is to be saved at the client side for future connections.
4. Simultaneously the data owner sends decryption scheme E^{-1} (.) to the data cloud for future distance decryption. It does not require that data owner should get involved in initial stage and can further reduce their involvement by handing over the task of decrypted indexing to the cloud.
5. Index in the cloud should again be encrypted by the owner's private key through any public key cryptography. During initialization, owner needs to forward their public key to the client who then recollects and decrypts the index from cloud.
6. In the course of PH , each time the client is required to go for index node which results node E(I) that computes the indexes , and are sent to the data cloud which decrypts and re-codes them for the client
7. It ensures that, only client can receive an encrypted version of the actual indexes that are acceptable and tolerable for any query processing. Whereas additionally to prevent the cloud from accessing the actual indexes after decryption, the client requires Private Key , prior to forwarding them to the cloud from accessing the actual indexes after decryption.
8. Text decryption scheme is already sent by the data owner to cloud and the decrypted indexes are encrypted by the text encryption scheme having at client end .
9. And finally entering the private key ,client will get the specified output for requested query.

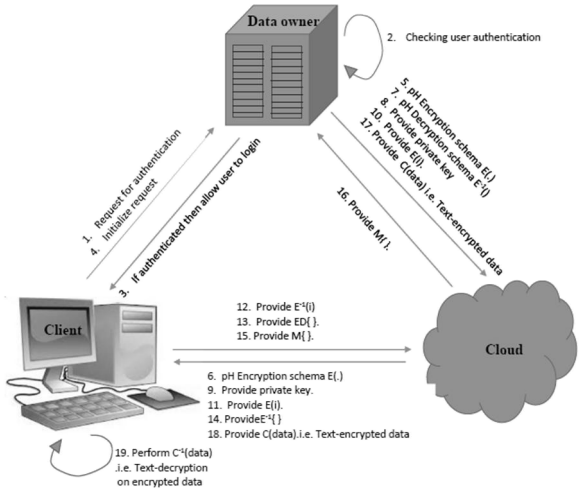

Fig. 3. Privacy-Preserving Query Processing Framework

7 PRIVACY-PRESERVING QUERY PROCESSING ALGORITHMS USED

A. pH Encryption algorithm

1) Start.

2) Take any number & multiply by 13 and store that answer.

3) Convert that answer into data type String and store into string variable fs.

4) Initialize integer array ak[ ] of size 10.

5) Initialize index counter variable to zero.int ak_ind=0;

6) For int i=0 to i<fs.length

Integer k = Integer.parseInt(fs.valueOf(fs.charAt(i)))+1;

If ((k==10) and (ak_ind>0))

ak[ak_ind-1]=ak[ak_ind-1]+1;

ak[ak_ind]=0;

Increment ak_ind by 1;

Else

ak[ak_ind]=k;

Increment ak_ind by 1;

End IF

End For

7) For int j=0 to j<ak_ind

enc_val=enc_val.concat(ak[j].toString());

End For

8) Return String s1.

9) Stop

B. pH Decryption algorithm

1) Start.

2) Take String s1.

3) Initialize int prime=13.

4) Convert String s1 into integer and store it in integer variable dc.

5) Initialize int dec=0;

6) dec=dc/prime.

7) Stop.

C. Text Encryption algorithm

1) Start.

2) Take a Text (i.e. fname , lname , email).

3) For int q=0 to q<Text.length

Ch[q] = (char) (text.charAt (q) +key);

End For.

4) Stop.

D. Text Decryption algorithm

1) Start.

2) Take an Encrypted text (i.e. efname , elname , eemail).

3) Calculate Text length.

4) For int q=0 to q<Text.length

Ch[q] = (char) (encryptedText.charAt (q) - key);

End For.

5) Stop.

8 SYSTEM MODEL FOR KNN ON R-TREE INDEX

Consider the following Fig-4, data owners may outsource their query services and data, but data is very sensitive and private assets of them and it should be protected from the service provider and the querying users in some extent. Data owner might be update, query and authorize access on the data, while the service providers in cloud should know nothing about especially detailed data about data, and query users should know not more than the exact answers for what she/he is querying[2].

On the other hand, query users need to query and exact data from cloud, but the query might disclose some sensitive information, behaviour patterns of the user. For example, when Bob searches a website, such as Face book, for friends who share the all general backgrounds things (e.g., age, education, home address) with her should not disclose the query that involves her own details to the cloud. Privacy of data owners and query users are defined as data privacy and user privacy respectively [10].

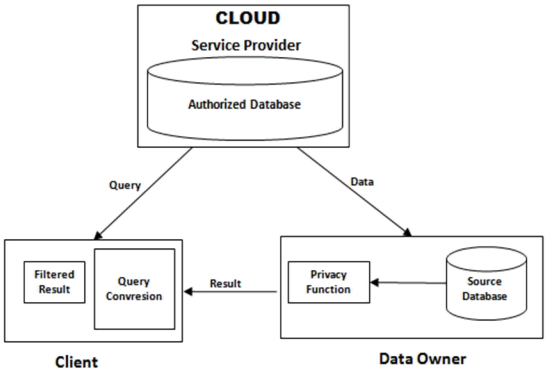

Fig.4. Architecture of Data Service on Cloud

It shows increasing importance as cloud computing in more businesses to outsource their data and various querying services. Hence, most of the study that, including how to outsource their data, how to make privacy on private data and how to retrieve the data by using appropriate query, the solution for all this problem is secure traversal framework and encryption scheme based on the privacy homomorphism. The framework is scalable to the large datasets by developing an index-based approach. Depending upon this framework, secure protocols such as k-nearest-neighbour queries (ken) on R-tree index are used. Highly Enhanced developing techniques are used to improve the efficiency of query processing protocols [2].

9 MATHEMATICAL MODEL

Let, Query is decrypted at cloud and encrypted at user level with using key.

M= (Q,Σ, δ, q_0, q_3)
Q= q0, q_1, q_2, q_3, q_3
q_0=initial stage
q_3=final state
Σ= (Iq, enc, dec, PH, Oq)

enc= Encryption key
dec = Decryption key
PH=Privacy Homomorphism
Iq= Input Set (input query)
Oq= Output Set (query result)

By using Euclidean distances: given two dimensional points $\vec{X}$ and $\vec{Y}$

The sum of square of distance in each dimension

$$\textbf{Dist}\ (\vec{X}, \vec{Y}) = \sum_{i=j}^{d} (X_i - Y_i)^2$$

I= (I_1, I_2……..I_n)
O=(O_1, O_2,.....O_1)

By using grammar rule,
δ (q_0 , input query) →q_1
δ (q_2 ,enc) →q_0
δ (q_0 , Privacy Homomorphism) →q_1
δ (q_2 ,dec) →q_1
δ (q_1 , query result) →q_3

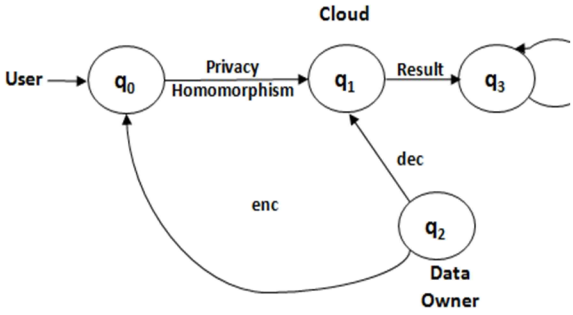

Fig. 5. Mathematical Models

10 EXPERIMENTAL RESULT

By analyzing the performance of our proposed Framework. While no existing work can be directly compared with it, the purpose is to show the feasibility and study its computation and query response time under various query types and parameter settings.

Table 1: Sample datasets and its value.

Parameter	Symbol	Value
R-tree Record	N	112,143
Page size	-	
Threshold for range query	τ	500-1500
k of kNN query	k	1-30

X Axis = τ
Y Axis = Response Time (ms)

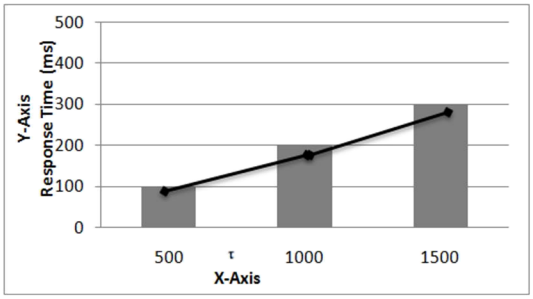

Fig. 6. Distance Range Query Performance

X Axis = k
Y Axis = Response Time (ms)

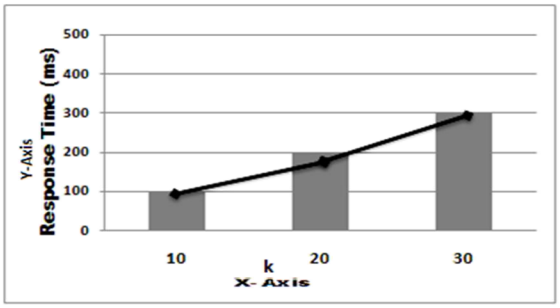

Fig. 7. Privacy Homomorphism Query Performance

As an expected result, the Distance Range Query shown in fig.6, increases moderately as τ increases. Similarly the query response time shown in fig.7 with using Privacy Homomorphism query, increases moderately as k increases.

11 CONCLUSION

As per the process mentioned herewith a study is conducted on processing problems of private queries on indexed data in a cloud. A secure traversal framework in indexed environment is given to secure protocols for such classic queries.

The assumptions and approached mentioned in this paper are thoroughly useful, efficient to perform and effectively used under settings of

different parameters. It has been summarized that the process mentioned here, on privacy homomorphism, is used to protect processing queries on cloud is high scalable.

12 REFERENCES

[1] Rupali s. khachane. Secure Query Processing of Outsourced Data Using Privacy Homomorphism: kNN and Distance Recoding Algorithm. International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064,2014.

[2] Guo, Yubin, et al. "A solution for privacy-preserving data manipulation and query on nosql database." Journal of Computers 8.6 (2013): 1427-1432.

[3] Hu, Haibo, et al. "Processing private queries over untrusted data cloud through privacy homomorphism." Data Engineering (ICDE), 2011 IEEE 27th International Conference on. IEEE, 2011.

[4] Nandhini, N., and P. G. Kathiravan. "An Efficient Retrieval of Encrypted Data in Cloud Computing."

[5] RakeshAgrawal, Jerry Kiernan, RamakrishnanSrikant, and YirongXu. Order preserving encryption for numeric data. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, SIGMOD '04, pages 563–574, New York, NY, USA, 2004. ACM.

[6] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[7] TingjianGe, Stanley B. Zdonik, and Stanley B. Zdonik. Answering aggregation queries in a secure system model. In VLDB, pages 519–530, 2007.

[8] HakanHacgm, BalaIyer, and SharadMehrotra. Efficient execution of aggregation queries over encrypted relational databases. In YoonJoon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, Database Systems forAdvanced Applications, volume 2973 of Lecture Notesin Computer Science, pages 125–136. Springer Berlin Heidelberg, 2004.

[9] Varghese, Jiss, and Lisha Varghese. "Homomorphic Encryption for Multi-keyword based Search and Retrieval over Encrypted Data."

AUTHOR PROFILES:

Dr. Pradeep .K. Deshmukh, He received PhD, M.E, B.E in Computer science & Engineering. His key research interest include Cloud computing, Network Security, ANN, Operating Systems. He is currently working as Professor in Department of Computer Engineering, Rajarshi Shahu College of Engineering Affiliated to Savitribai Phule Pune University India (M.S) Having total experience of about 23 years.

Rupali S. Khachane had completed Bachelor of Engineering in Information Technology and Masters in Engineering in Computers, from RajarshiShahu College of Engineering, Tathawade, Pune-33 India (M.S) under Savitribai Phule University Pune, MH, India.