



Secure kNN Query Processing in Untrusted Cloud Environments: an Overview

Monika D.Rokade¹, S.A.Kahate², K.S..Kore³

M.E. Student, Department of Computer Engineering, SPCOE, Dumberwadi, Pune, Maharashtra, India¹

Head, Department of Computer Engineering, SPCOE, Dumberwadi, Pune, Maharashtra, India²

Department of Computer Engineering, SPCOE, Dumberwadi, Pune, Maharashtra, India³

ABSTRACT: Nowadays, data are stored to a third party in cloud environments and query processing is also done by the third party to reduce the expense to maintain the system. Although there are lots of advantages in using independent third parties in query processing, security problems become more crucial since we cannot completely trust the third parties which can be easily corrupted or malfunctioning. The security problems with untrusted third parties are multifaceted in several areas such as privacy, authentication, and recovery. For privacy, the third party should not be able to know what the user's query is since the query itself describes the user's interest. For authentication, the user should be able to verify that the information from the third party is not tampered since the correctness of the query results depends upon the correctness of the information from the third party. For recovery, when the result is found to be forged by an adversary, we should be able to find the adversary and get a correct result by removing the adversary. To address these challenges, we propose several schemes. First, with respect to secure kNN query processing and secure proximity detection, we give novel schemes based on Mutable Order Preserving Encryption (MOPE) and Secure Point Evaluation Method (SPEM). Second, for authenticated top-k aggregation, we suggest novel schemes using Three Phase Uniform Threshold Algorithm, Merkle Hash Tree, and Condensed-RSA. Third, for detecting malicious nodes, we propose novel algorithms based on Additively Homomorphic Encryption and Multipath Transmission. Our experimental evaluation and security analyses demonstrate that robust mechanisms can be deployed with a minimal amount of computational and communicational expense.

I. 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, the client, and the cloud refer Fig- 1. The social networking is one of the rising sectors facing such type of privacy problem [2]. Cloud Computing is new platform to deploying, managing, and providing solution to the various types of storage, platform problems using internet-based processing. However, it is very sensitive issue to upload our personal data on the cloud because data privacy is the big issue and major problem of security. Sensitive information has to be encrypted before outsourcing, which creates the effective infrastructure. The services such as Goggle Docs, Amazon EC2, Microsoft Azure, and Online file storage etc. are the examples of cloud computing and they are widely used by many people worldwide. data utilization services and that is really 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]. For the privacy of the data, various general solution in recently research papers are deposited to show study on the data privacy, the most general solution in recently done research papers are encryption. It means data deposited service provider must be encrypted to avoid information leakage on the cloud. Agrawal et al [4] 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 rewritten 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's query completely and providing strong privacy



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

and confidentiality, query anonymisation usually uses k-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 Wenyang Bai discussed how to enforce data privacy and user privacy over outsourced database service in [8]. Hu et al. [9] proposed one of the solution based on secure traversal framework and privacy homomorphism based encryption scheme. And secure protocols for processing k-nearest-neighbor queries (kNN) on R-tree index is given. In the authors following work [7], they integrated indexing techniques with secure multiparty computation (SMC) based protocols to construct a secure index traversal framework. In this framework, the service provider cannot trace the index traversal path of a query during evaluation, and hence keep privacy of users. Their protocols for query are complex, and hard to implement. 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 researchers 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, and 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 [12]. Our approach may also apply to protect query privacy in outsource scenarios

II. ARCHITECTURE OF A PROPOSED WORK

Consider a data management system hosting data service, as illustrated in Fig-ure 2, in which three different entities are involved: data owner, data user and a storage server. 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, $C = \{f_1, f_2, \dots, f_n\}$ 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[10].

We rely on the homomorphic encryption [11] to provide a strong privacy protection fo the sensitive data. Homomorphic encryption allows addition and multiplication without the need for decryption to be directly performed on cipher texts and that too without loss of generality. We use the popular Paillier's homomorphic encryption here [11].

Proposed Work

A lot of studies were done prior to this which provides a secure framework and substitute encryption schemes, both are imperative. Here, we wish to present a systematic and result oriented solution based on Privacy Homomorphism (PH). It is an encrypted transformations mapping set of operations on plain text to another set of operations on Ciphertext[12]. There are three basic steps to solve problems of outsourced data processing query in cloud and for client. 1. An index 2. consists of multiple nodes which are used for processing queries including traversing the nodes. Data owner and cloud may not be able to trace the access pattern hence hardly get any clue of the query. Here, a client-lead processing terms excludes the display of query to third party. To assess various types of complex queries 3. such as kNN and other distance-based queries, an inclusionary set of client-cloud protocols must be organized to work together with a PH to supports most arithmetic operations. Provide security and evaluate complexity of the proposed algorithms and protocols. Particularly various optimization techniques help to increase the protocol efficiency to show x their privacy benefits.

III. 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,

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

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
2. During this process data owner sends an encrypted variant of index –E (I). In each index node, the key entry e.g. e1, e2, e3 is encrypted by encryption scheme E(·),
3. Although the pointers e.g., p1, p2, p3 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-I(.)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 traversal, each time the client is required to go for index node which results node E(I) that computes the local distances, and are sent to the data cloud which decrypts and re-codes them for the client
7. This re-coding ensure that, only client can receive an encrypted version of the actual distances that acceptable and tolerable for the query processing. Whereas additionally to prevent the cloud from accessing the actual distances after decryption, the client is required to scramble local distances prior to forwarding them to the cloud from accessing the actual distance after decryption. 8. The traversal begins at the root node, and the node access process repeats until the query is completed.

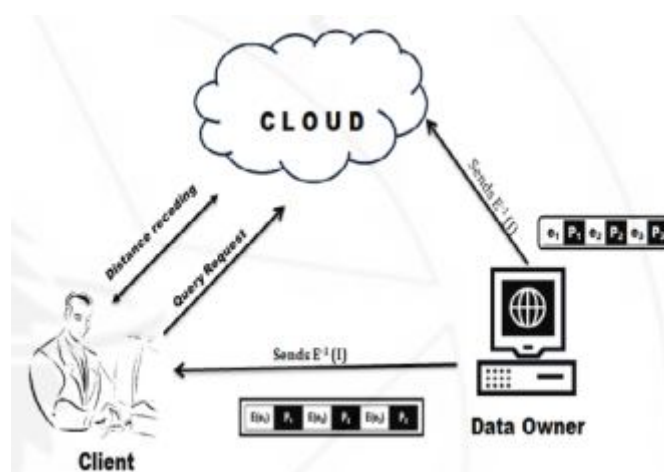


Fig. 1 proposed System framework

Privacy-Preserving Query Processing Algorithms Used

Distance-based Queries

- a) Owner sends Encrypted index E(I) to the client
- b) Owner sends the decrypted E-I(I) to the cloud
- c) Client initializes the root of E(I) as I , the next node to access
- d) Client retrieves index node(i), computes and scrambles the local distances
- e) Cloud receives the scrambled local distances, decrypts and recodes them.
- f) Client updates the query and move to the next node (follow all above steps)

Distance Recoding Scheme

- a) Local distances computed as per above, are encrypted by E(·)
- b) Sent them to cloud for decryption.
- c) The client scrambles the encrypted distances and the cloud decrypts them.
- d) Instead of forwarding the sign results directly, the



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

cloud must encrypt the distances to prevent the client from accessing the actual distances, the process called is distance recoding where it sends back a recoded version of the distances that are only sufficient for distance comparison.

IV. BASIC PRIVATE QUERY PROCESSING

One of the main challenges for private query processing is to privately represent a given user query, and find and retrieve the qualified values from Rpub. for the query. In our basic framework, we propose to use a novel approach of data bucketization with homomorphic encryption to solve this challenge, and we provide perfect privacy of query in distinguish ability for clients, meaning that the adversaries who may have control of servers should not be able to differentiate accesses of different queries on Rpub. B. One advantage of our framework over any other PIR protocols is that our framework can answer a query in only one round of client server interaction, thus saving the bandwidth for the server [12]. Figure 5: BHE. In this protocol, before processing any queries Steps

- 0.1) Server sends the bucket summary S of its database to the client
- 0.2) Client sends her public key K_{pub} to the server. Then to process a query q .
1. Client formulates an encrypted query vector Q' based on S and q , and sends Q to the server.
2. Server performs blind processing on Q' and public database, sends the answer vector V back to the client
3. Finally, the client decrypts V and reconstructs the answer to the query q .

Security Domain

Security conditions are checked and analyzed from the client and cloud/ data owner angle. Initially, data security of the proposed framework depending upon theoretical results from PH are shown [12], afterwards, understands the query security especially the security of scrambling process and the optimization for distance re-coding.

Data Security

It has been based on two factors - the security of the secret keys in the PH and distance recoding scheme.

1) Key Security:

PH security is depend upon the encryption and decryption of key against the oppose of set number of ciphertext [12]

2) Distance Recoding:

Scrambled modified distances are unable to react. It means they are independent.

V. QUERY PRIVACY

Query Security is based on two factors. The security of the scrambling and “untraceable root access”, latter means cloud is unable to point out or short list the query when first node accessing. Cloud continues to treat it as root node. 1) Scrambling Security: Arithmetic operations are used to derive deviations on the basis of initial seeds and composite seeds because composite seeds are in large volume to derive at few steps. 2) Untraceable Root Access: It has been observed that scrambling process is quiet effective and trustworthy which helps convert genuine distances into relative ones. This decomposes substitute entries because cloud cannot narrow the query in the root access.

VI. 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.

REFERENCES

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



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

- [3] Nandhini, N., and P. G. Kathiravan. "An Efficient Retrieval of Encrypted Data In Cloud Computing."
- [4] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 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.
- [5] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.
- [6] Tingjian Ge, Stanley B. Zdonik, and Stanley B. Zdonik. Answering aggregation queries in a secure system model. In VLDB, pages 519–530, 2007.
- [7] Haibo Hu and Jianliang Xu. Non-exposure location anonymity. In Yannis E. Ioannidis, Dik Lun Lee, and Raymond T. Ng, editors, ICDE, pages 1120–1131. IEEE, 2009.
- [8] Yonghong Yu and Wenyang Bai. Enforcing data privacy and user privacy over outsourced database service. JSW, 6(3):404–412, 2011.
- [9] Hakan Hacgim, Balajyer, and Sharad Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In Yoon Joon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, Database Systems for Advanced Applications, volume 2973 of Lecture Notes in Computer Science, pages 125–136. Springer Berlin Heidelberg, 2004.
- [10] Varghese, Jiss, and Lisha Varghese. "Homomorphic Encryption for Multi-keyword based Search and Retrieval over Encrypted Data."
- [11] C. Gentry. Fully homomorphic encryption using ideal lattices. In STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing, pages 169–178, 2009.
- [12] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In Proc. 5th International Conference on Information Security, 2002.