International Journal of Engineering Research And Advanced Technology (IJERAT)
ISSN: 2454-6135 [Volume. 02 Issue.03, March-2016]

A Secure Framework for efficient search and mining over an encrypted data using homomorphic encryption

N.Nivethitha[1], D.Raghu Raman[2]

Department of CSE

IFET College of Engineering

Gangarampalayam,villupuram

India

Abstract*:*

In the emerging networked environments learning tasks are encountering situations in which the relevant data exists in a number of geo-graphically distributed databases that are connected by communication networks. For the past decade with recent esteem of cloud computing, user have the opportunity to outsource their data .Since the data on cloud is in encrypted form performing data mining task as the challenging issue and cloud user have privacy and security issue in protecting both data and query privacy among data owner, client and the cloud. In this paper we focus on secure retrieving of an outsourced encrypted data without loss of generality and provides privacy and security by assigning attribute key to the data. Further we enhance a multi level trust privacy preserving scheme to provide a solution based on secure traversal index based framework and homomorphic encryption scheme for processing queries on R-Tree index.

Keywords- *Security, Privacy homomorphism's, R-tree index traversal framework, outsourced data, loss of generality.*

I.INTRODUCTION

Recently Cloud computing provides a way for the cloud user to outsource their data, which is useful for both individuals and organization. Due to this the outsourced data disclose some sensitive information which needs to be encrypted before outsourcing. An Irrespective of the primary encryption scheme, mining task over an encrypted data becomes very challenging without decrypting.

To protect a user record only when that record is a part of mining process it address some of the privacy/Security requirements such as discretion of the encrypted data, discretion of a user's query record and defeat data access patterns. The social networking is one of the rising sector facing such type of privacy problem. Cloud Computing is new platform to deploying, managing, and providing solution to the various types of storage, platform problems using internet-based infrastructure.

The services such as Goggle Docs, Amazon EC2,Microsoft Azure which are used in worldwide. In the existing work data should be retrieved only from the classification based on k-nn classifier. It takes high computation complexity for classify each user record and to provide security issues and also duplication of the data. It address some

of the issues such as intermediate k-nn in classification not to be disclosed to cloud, very tricky to find majority class label surrounded by the neighbors and did not address the access pattern issue during query processing on encrypted data. Many intermediate computation to be performed based on non-encrypted data and do not possess semantic security.

Specifically We focus on the secure retrieval of an outsourced encrypted data by providing attribute Key to that data, then overall mining will be efficient and more secure search with less computational complexity. So we further enhancing Multi-level Trust privacy preserving with semi honest model. This Proposed scheme provides solution using traversal framework and homomorphic encryption for processing queries on R-tree index. By using this technique it does support classification over encrypted data by providing attribute key to the data. By implementing this method overall mining will be efficient and more secure search with better classification with less computational complexity.

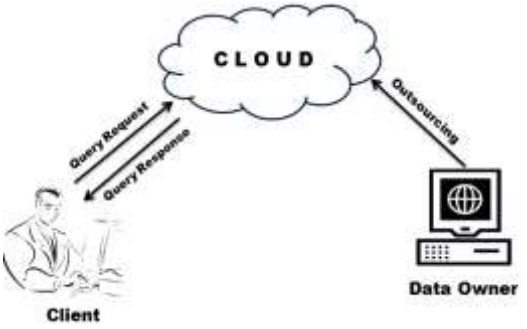


Fig 1. General Model for query processing in Cloud

However it is very sensitive to upload a personal data on the cloud because data privacy is the big issue and major problem of security. Before outsourcing the data sensitive information needs to be encrypted which creates the valuable utilization of data and is really big challenging task. Our approach may also apply to protect query privacy in outsourced scenarios.

2. ARCHITECTURE OF A PROPOSED WORK

Consider system hosting service of data, as illustrated in Figure, in which three different entities are involved as proprietor, enjoyer and a server. The Proprietor has a collection of data files expectant to outsource their data from home systems to universal space for great flexibility. To enable search and retrieving an encrypted data is of overriding importance. The 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.

We rely on the homomorphic encryption scheme to provide a sturdy privacy fortification of 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.

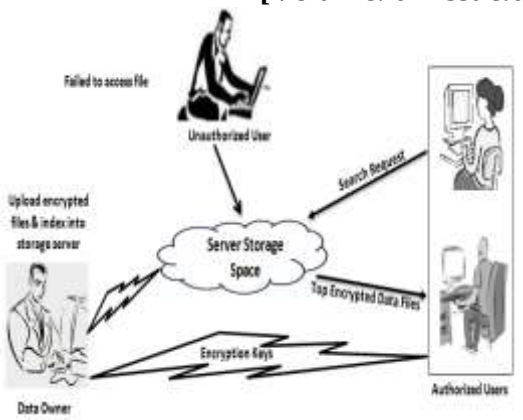


Fig 2. Search and access an outsourced encrypted data

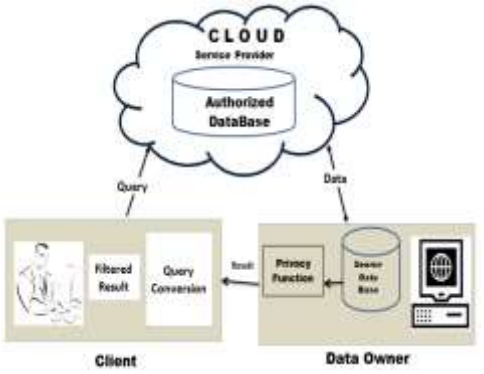


Fig 3. System Model for Index Framework

Consider the following Fig-3, proprietor has a collection of data and stored in authorized database.In that data it enclose some sensitive information that needs to be encrypted and to prevent it from unauthorized user. When Query user needs to retrieve a data first they send a request to data owner and provide an username and password ,after that they able to give a query. Then data to fetched based on index based traversal and to provide attribute key for the requested file then it to be retrieved by the query 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. This framework is scalable to the large data sets by developing an index-based approach. Depending upon this framework, secure protocols for processing queries on R-tree index are used.

3. RELATED WORK

We discusses various technique focused on secure retrieval of outsourced encrypted data that stored in cloud server. We also identify their strength and weakness. In rest of paper we demonstrate how our scheme overcome those weakness.

Secure multidimensional range queries over outsourced data are explored in [21-22] over encrypted cloud data. It has the problem of supporting multidimensional range queries on encrypted data. The problem is motivated by secure data

outsourcing applications where a client may store his/her data on a remote server in encrypted form and want to execute queries using server's computational capabilities.

A bucketization procedure is used for provides answering multidimensional range queries on multidimensional data. For a given bucketization scheme, we derive cost and disclosure-risk metrics that estimate client's computational overhead and disclosure risk respectively.

Given a multidimensional dataset, its bucketization is posed as an optimization problem where the goal is to minimize the risk of disclosure while keeping query cost (client's computational overhead) below a certain user-specified threshold value. The major issue is computational overhead and returned set of record contain some false positives. Managing and accessing data in the cloud: Privacy risks and approaches explored in [2],[9],[10]. In this paper, characterize different aspects of the privacy problem in emerging scenarios. It will illustrate risks, solutions, and open problems related to ensuring privacy of users accessing services or resources, sensitive information stored at external parties, and accesses to such information. Ensuring proper privacy and protection of the information stored, communicated, processed, and disseminated in the cloud as well as of the users accessing such information is one of the grand challenges of our modern society. The major issue is Introduce novel privacy risks of improper information disclosure and dissemination. Implementing gentry's fully-homomorphic encryption scheme explored in [7], [15]. Smart and Vercauteren implemented the underlying "somewhat homomorphic" scheme, but were not able to implement the bootstrapping functionality that is needed to get the complete scheme to work. We show a number of optimizations that allow us to implement all aspects of the scheme, including the bootstrapping functionality. Our main optimization is a key-generation method for the underlying somewhat homomorphic encryption, that does not require full polynomial inversion. The advantage is to reduce asymptotic complexity, when working with dimension-*n* lattices. The issue arised here is Privacy issues for the outsourced encrypted data and Space/time trade-off is arised for fully- homomorphic scheme.

4.MODULE IN PROPOSED WORK

4.1 User registration

Used to assigns member ID, member Name, password, then stores the necessary information into the database. A user's permission should be distributed through a secure channel. Data sharing products, our system relies on password verification for authenticating users. To further improve the security some captures are used.

4.2 User authorizer

In this module, authorizer will login and provide access permission to data owner. After providing authorization to data owner, the data owner could upload the files into database and take corresponding action.

4.3 Data owner

The purpose of this module is to maintain a profile of the query user and to view the request from them. once user is registered then data owner provide ID password, Secret key. assigns a doc ID for this document and stores the encrypted data in the path file Path, then inserts a record into the table docs. In addition, the owner can encrypt the keys using his/her private key and store them into the table docs.

4.4 File verifier

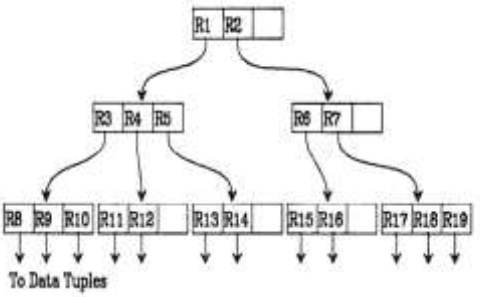
In this module, file verifier will login and verify the uploaded files. After uploading data owner file, verifier will assign security encryption key to all the files in database.

4.5 Query user/data miner

Query users need to provide query and extract data from database, but it might reveal some susceptible information, behaviour patterns of the user. Query user will send request to extract file from the database. Query user will get the corresponding requested file by receiving secret key from data owner.

5. R-TREE INDEX ALGORITHM

1. Every leaf node contains between m and M index record unless it is root.
2. For each tuple index record(I,tuple identifier) in a leaf node, I is the smallest rectangle that spatially contains the n-dimensional data object represented by the indicated tuple.
3. Every non-leaf node has between m and M children unless it is a root.For each entry (I,child-pointer) in non-leaf node, I is the smallest rectangle that spatially contains the rectanlgle in the child node.
4. The root node has at least two children unless it is a leaf.
5. All leaves appear on the same level.



5.1 Algorithm search:

1.In an R-Tree root node is T, find all index records whose rectangles be related a search rectangle S. S1 (search subtrees) if T is not a leaf, Check each entry E to determine whether EI overlaps S. For all overlapping entries, invoke search on the tree whose root node is pointed by Ep.

2.S2 (search leaf node) if T is a leaf, check all entries E to determine whether EI overlaps S if so E is a qualifying record.

5.2 Homomorphic Encryption Algorithm:

The security aspects, definitions, and models of homomorphic cryptosystems are the same as those for other cryptosystems. This algorithm comes under the RSA cryptography scheme. It consist of 512 bytes of data.

consent to the message space (M, o) is finite group, and σ be the security parameter. A homomorphic private-key encryption scheme on Misaquadruple (K, E, D, A) of probabilistic following functionalities:

- **Key Generation:** On input 1^{σ} the algorithm K outputs an encryption/decryption key pair $(k_e, k_d) = k \in \boldsymbol{K}$, where $\boldsymbol{K}$ denotes the key space.
- **Encryption:** On inputs $1^{\sigma}, k_e$, and an element $m \in \boldsymbol{M}$ the encryption algorithm E outputs a ciphertext $c \in \boldsymbol{C}$, where $\boldsymbol{C}$ denotes the ciphertext space.
- **Decryption:** The decryption algorithm D is deterministic. On inputs $1^{\sigma}, k$, and an element $c \in \boldsymbol{C}$ it outputs an element in the message space $\boldsymbol{M}$ so that for all $m \in \boldsymbol{M}$ it holds: if $c = E(1^{\sigma}, k_e, m)$ then $Prob[D(1^{\sigma}, k, c) \neq m]$ is negligible, i.e., it holds that $Prob[D(1^{\sigma}, k, c) \neq m] \leq 2^{-\sigma}$.
- **Homomorphic Property:** A is an algorithm that on inputs $1^{\sigma}, k_e$, and elements $c_1, c_2 \in \boldsymbol{C}$ outputs an element $c_3 \in \boldsymbol{C}$ so that for all $m_1, m_2 \in \boldsymbol{M}$ it holds: if $m_3 = m_1 \, o \, m_2$ and $c_1 = E(1^{\sigma}, k_e, m_1)$, and $c_2 = E(1^{\sigma}, k_e, m_2)$, then $Prob\left[D\left(A(1^{\sigma}, k_e, c_1, c_2)\right)\right] \neq m_3]$ is negligible.

6. EXPERIMENTAL RESULTS

7.CONCLUSION

To Protect a privacy/security various classification techniques have been proposed over the past. The existing techniques is not applicable to retrieve an outsourced encrypted data using k-nn classification where the data gets duplicated and not produce accurate mining resuls. This paper proposed a secure traversal framework for searching and retrieve the data using homomorphic encryption scheme for ensuring the confidentiality of processed data. Also We provide a secure and more efficient solution to the classification problem in our future work.

REFERENCE

[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.

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

[4] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, 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.

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

[6] TingjianGe, 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 JianliangXu. Non-exposure

location anonymity. In Yannis E. Ioannidis, DikLun Lee, and Raymond T. Ng, editors, ICDE, pages 1120–1131. IEEE,2009.

[8] Yonghong Yu and WenyangBai. Enforcing data privacy and user privacy over outsourced database service. JSW, 6(3):404–412, 2011.

[9] Hakan Hacgm, BalaIyer, 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 forAdvanced Applications, volume 2973 of Lecture Notesin 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."