Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.507 – 512

RESEARCH ARTICLE

SECURED SEARCHING OF VALUABLE DATA IN A METRIC SPACE BASED ON SIMILARITY MEASURE

R. Rathika[1], Dr. K. Raja[2]
[1]P.G Student, M.E CSE, Alpha College of Engg, Chennai, T.N, India
[2]Dean (Academics) Alpha College of Engg, Chennai, T.N, India

Abstract— The aim of the project is collecting the similarity queries from various users and stored in the database. In this paper, we are mainly concentrating on privacy only. Here data owner, service provider, trusted clients are used. Here it is able to maintain data confidentiality with respect to untrusted parties including the service provider. Data owner and service provider and trusted client are used. Data owner is one who stores the data in the database. Here service provider is the third party who maintains the data in the database. Trusted client is one who needs the data from the database. In this paper the data owner provide the privacy to the sensitive information .Here I took medical related information so I collected the medical related data and stored in my database such as fever, headache and diabetes disease related information. Here all the data's are stored in the hierarchical order in a subject wise or age wise or disease wise. The cloud computing setting in which similarity querying of metric data is outsourced to a service provider. Only authorized users are allowed to access the data. Nobody else including the service provider should be able to view the data. So that data will be kept private. Based on the queries it will be revealed to the trusted users alone. This transformation technique offers perfect data privacy for the data owner but it gives the final result at multiple rounds of communication. This technique also provides an interesting trade-off between query cost and accuracy. Existing solutions either offer query efficiency at no privacy or they offer complete data privacy while sacrificing query efficiency. But the proposed methods are very secure and efficient.

Key Terms: - query processing; security; integrity; protection; data owner; service provider; trusted client

I. INTRODUCTION

Cloud computing services enable individuals and organizations to outsource the management of their data with ease and at low cost, even if they lack IT expertise. Cloud computing enables scalability with respect to storage and computational resources as the number of service requests grows, without the need for costly investments in hardware and maintenance. Consider the example of a real-estate company that owns a large database with descriptions of properties and their locations. The company (i.e., the private data owner) wishes to allow authorized users (e.g., paying customers) to query for properties situated within a certain geographical region. To save on hardware investments and maintenance costs, the data owner outsources the management of its dataset to a service provider (SP) [4] that specializes in data storage and query processing. However, the SP may not be fully trusted, and could sell the data to a competitor. Furthermore, even if the SP is trusted, a malicious attacker can compromise the SP and gain unauthorized access to the data. To prevent such attacks, the data owner first encrypts the dataset according to a secret transformation and then uploads the encrypted data to the SP [4]. Only authorized users who know the transformation are able to learn the property locations.

II. RELATED WORK AND EXISTING MODEL

This paper focuses on the outsourcing of metric datasets [2]. The main objective is to enforce the user authorization specified by the data owner, even when the service provider cannot be trusted. It presents

techniques that protect location data from attackers, while allowing authorized users to issue queries that are executed efficiently by the SP [4]. In the literature, a number of concepts for securing databases have been studied. Private information retrieval technique [2]. In that technique gives query efficiency but never give the query privacy. Sometimes gives the query privacy while sacrificing the query efficiency and sometimes hide user query to the data owner. If user ask any query, in that technique just searching the query but never retrieve the query to the responding user. So there is no guarantee for retrieval and accuracy. It cannot prevent an attacker illegally copying the data from the data set.

Typically, cloud computing [1] providers attempt to solve any problem by offering the solutions only given to the data owner and are not to release outsourced data [2] to third parties. Nevertheless, even if the provider can compromise by anyone, the data is not guaranteed to be safe. Unintended leaks of data are reported regularly, and hackers may still exploit vulnerabilities to gain access to data. Therefore to believe that data owners will find it attractive to outsource encrypted rather than plain data.

The main related work is to first introduce existing work on indexing and nearest neighbour search techniques for metric data. Then will cover work on privacy and security of outsourced data. In the field of privacy-preserving data mining, perturbation techniques have been developed for introducing noise into the data, before sending them to the service provider. However, such an approach does not guarantee the exact retrieval of results. The idea of outsourcing database services to a service provider [4] was introduced by Hacigumus. Since then, various techniques have been developed to maintain the confidentiality of outsourced data.

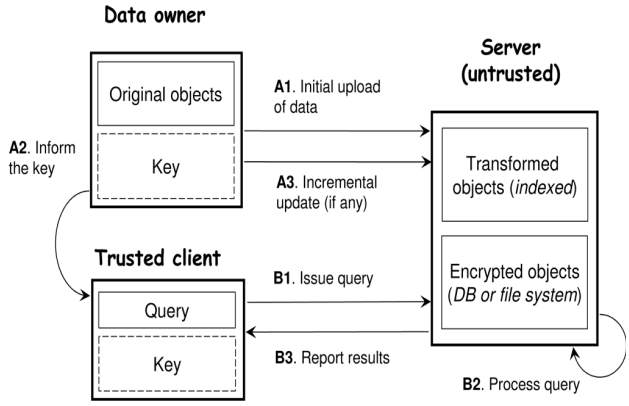

Fig. 1. Scenario overview

It consists of three entities: a data owner, a trusted query user, and an untrusted server. On the one hand, the data owner wishes to upload his data to the server so that users are able to execute queries on those data. On the other hand, the data owner trusts only the users and nobody else (including the server). The data owner has a set P of (original) objects and a key to be used for transformation. First, the data owner applies a transformation function (with a key) to convert P into a set P0 of transformed objects, and uploads the set P0 to the server (see step A1 in the fig). The server builds an index structure on the set P0 in order to facilitate efficient search. In addition, the data owner applies a standard encryption method (e.g., AES) on the set of original objects; the resulting encrypted objects (with their IDs) are uploaded to the server and stored in a relational table (or in the file system).Next, the data owner informs every user of the transformation key (see step A2). In the future, the data owner is allowed to perform incremental insertion/deletion of objects (see step A3). At query time, a trusted user applies the transformation function (with a key) to the query and then sends the transformed query to the server (see step B1). Then, the server processes the query (see step B2), and reports the results back to the user (see step B3). Eventually, the user decodes the retrieved results back into the actual results. Observe that these results contain only the IDs of the actual objects. The user may optionally request the server to return the actual objects that correspond to the above result set.

To use the term object for the metric data[2] of interest to the data owner. A transformed object then refers to an object obtained from a transformation. This brute-force solution is the one we mentioned in the Introduction. In the offline construction phase, the data owner applies the conventional encryption (e.g., AES) on each object and then uploads those encrypted objects to the server. At query time, the user needs to download all encrypted

objects from the server, decrypt them and then compute the actual result. As mentioned, it is perfectly secure, but its query and communication cost are both prohibitively high, and pay-as-you-go is not supported. This anonymization-based solution achieves data privacy by means of k-anonymity—the objects are generalized in such a way that each generalized object cannot be distinguished from k _ 1 other generalized objects. In the context of similarity search, it is able to confuse the ranking of transformed objects because k _ 1 of them have the same rank as the transformed object of the actual nearest neighbour. Thus, we still consider this solution as a competitor, even though k-anonymity is not a suitable privacy guarantee for our applications.

Existing solutions either offer query efficiency at no privacy or they offer complete data privacy while sacrificing query efficiency. But the proposed methods give a privacy and security. It's very secure and efficient. This technique having lot of disadvantages to overcome this problem we go for proposal.

III. ENCRYPTED HIERARCHICAL INDEX BASED SEARCH

Encrypted hierarchical index search , this module is always gives a privacy with query efficiency and query is guarantee to be accurate. my database only have medical related data such as fever, headache and diabetes disease related information. Here all the data's are stored in the hierarchical order in a subject wise or age wise or disease wise .For example two or more people affected by fever so they are asking fever related queries to the data owner. Then the data owner goes for similarity searching operation with help of the EHI algorithm. In First step is searching for fever related queries is available in the database or not. Here indexing is very important it is mainly used for efficient searching of data in the database. The searching operation is finished successfully. Then goes for fetching operation. The fever related queries [7] available in the database so the data is fetched and finally retrieve the data to the trusted users alone.

This method gives the final result at multiple rounds of communication during query processing. However, no solutions were proposed for the NN[9] query on those encrypted indexes. They capture various trade-offs among data privacy and query cost and accuracy [1][2]. Here transformation key, query processing and incremental data update are used.

The transformation key is used for client to verifying the secret key. The transformation key of EHI is simply an encryption key CK for standard encryption algorithm (e.g., AES).

The query processing technique is used to classify the query and send to the trusted client. It is something but retrieval of information from a database according to a set of retrieval criteria and the data itself remaining unchanged. Since the tree index stored at the server is encrypted, the server cannot process the NN[9] query by itself. An algorithm for communication between the client and the server needs to be developed in order to answer the NN[9] query correctly. The total response time of the algorithm consists of the round trip latency and the data transfer time. Round trip latency is also called round trip time. It is nothing but the length of the time taken for a sending a query to travel from a source to a specific destination and back again. Data transfer time is the time taken for a data transfer between drives and host system. The time is depending on the size of the data transfer and rate at which it can be transmitted to/from the host. These two measures are analogous to the seek time and transfer time in hard disks. Traditional best-first NN[9] search algorithms guarantee that the data transfer time is minimized. However, in the above context, they need to send a message to the server each time a node is requested. This would incur very high round trip latency.

The incremental data update is used for the data owner is able to insert and delete objects from the encrypted tree, similar to the way of using the underlying index [3].

IV. METRIC PRESERVING TRANSFORMATION

The same EHI type of operation is also done here(searching,indexing,fetching,retrieving).Metric preserving transformation, for evaluating the NN [9] query, after that MPT gives the final result at two rounds of communication during the query phase. Here we use distance bounding phase and candidate retrieval phase by using that two phases it gives the final result at single rounds of communication. Distance bounding phase main function is to filter the keyword in the database list and candidate retrieval phase is also filter the number in the database list. Here both are using the optimization method is mainly used for reducing the processing overhead and increasing the efficiency. How to reduce the processing overhead, first step is largest database split up into smallest database finally merge the database we get the result in single rounds.

The transformation key consists of an encryption Key CK, an integer A, and A pairs of the form (a_i,r_i) where a_i is an(anchor) object and r_i is a distance value.

The query processing technique is used to classify the query and send to the trusted client. It is something but retrieval of information from a database according to a set of retrieval criteria and the data itself remaining unchanged. Their order preserving encrypted values using OPE[2] With these encrypted distances, the following lemma states when an object p cannot become the NN [9] of q. In order to guarantee exact NN retrieval, the client needs to issue a distance range request to the server so that any object p satisfying OPE is retrieved.

The incremental data update is to maintain the data owner can perform the split the largest partition into two equal-sized partitions. Merge the smallest partition with its nearest partition. The maintenance overhead is small because it involves only three partitions.

V. FLEXIBLE DISTANCE-BASED DYNAMIC HASHING

Here we propose A new hashing-based technique is called flexible distance-based dynamic hashing, for processing the NN [9] query. The main advantage of this technique is that the server always returns a constant-sized candidate set. Candidate set is nothing but total number of item set we are used in our transaction.

The client then refines the candidate set to obtain the final result. Even though FDH is not guaranteed to return the exact result, the final result is very close to the actual NN in practice. During query processing, FDH allows the client to specify an integer parameter Θ for increasing the accuracy of a query result, without rebuilding the transformed data stored at the server. In addition, our FDH method employs a novel technique for conceptually linking similar [3] hash buckets, in order to maximize the utility of the transformed data for answering queries.

The transformation key consists of an encryption Key CK, an integer A, and A pairs of the form (a_i,r_i) where a_i is an object and r_i is a distance value.

The query processing strategy is to apply a similarity search on the above metric space index .The pseudocode of the searching algorithm for FDH. The client specifies an additional integer parameter and requests the server to retrieve the tuples whose bitmaps are the closest to the query bitmap BM. After receiving the result tuples from the server, the client decodes them into original objects and computes their distances from q.

The client refines the candidate set to obtain the final result. The FDH is not guaranteed to return the exact result but the final result is very close to the actual result. This parameter provides a trade-off between the query cost and accuracy [2]. It always gives flexibility to the user. The FDH gives a final result at single rounds of communication.

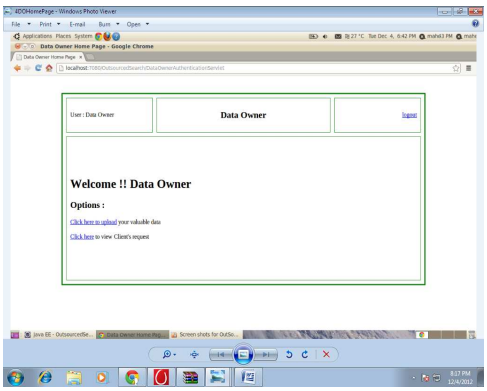

Fig. 2. Data owner login page

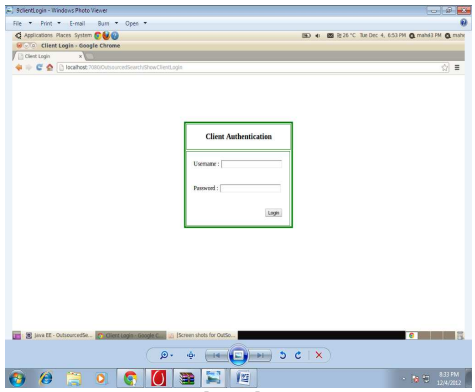

Fig. 3. Client login page

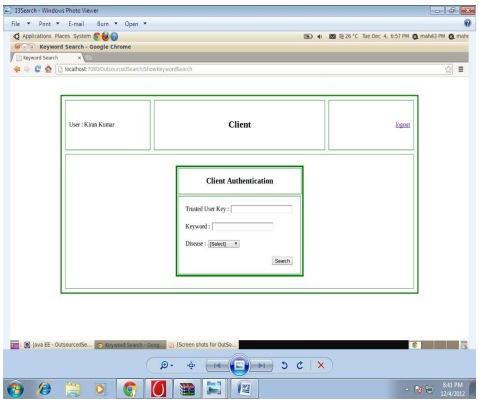

Fig. 4. Search page

VI. CONCLUSIONS

Existing solutions either offer query efficiency at no privacy, or they offer complete data privacy while sacrificing query efficiency. It is attractive to be able to maintain data confidentiality with respect to untrusted parties, including the service provider. The paper presents methods to encode a dataset such that only authorized users can access the content, while the service provider "blindly" evaluates queries [4], without seeing the actual data. It is important for the data owner to choose an appropriate transformation method that best matches the requirements. We are proposing three transformation methods. The first method is encrypted hierarchical index search algorithms gives the final result multiple rounds of communication. The second method is Metric Preserving Transformation method guarantees correctness of the final search result, but at the cost of two rounds of communication. The third proposed method is Flexible Distance-based Hashing methods finishes in just a single round of communication, but does not guarantee retrieval of the exact result. But actual result is very close to the exact result. This transformation methods achieve different trade-offs between the data privacy and query efficiency.

REFERENCES

[1] T. Bozkaya and Z.M. O ¨ zsoyoglu, "Indexing Large Metric Spaces for Similarity Search Queries," ACM Trans. Database Systems, vol. 24, no. 3, pp. 361-404, 1999.

[2] M.L. Yiu, I. Assent, C.S. Jensen, and P. Kalnis, "Outsourced Similarity Search on Metric Data Assets," DB Technical Report TR-28, Aalborg Univ., 2010.

[3] M.L. Yiu, G. Ghinita, C.S. Jensen, and P. Kalnis, "Outsourcing Search Services on Private Spatial Data," Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 1140-1143, 2009.

[4] G.R. Hjaltason and H. Samet, "Index-Driven Similarity Search in Metric Spaces," ACM Trans. Database Systems, vol. 28, no. 4, pp. 517-580, 2003.

[5] P. Ciaccia, M. Patella, and P. Zezula, "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces," Proc. Very Large Databases (VLDB), pp. 426-435, 1997.

[6] E. Damiani, S. D. C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing Confidentiality and

Efficiency in Untrusted Relational DBMSs. In CCS, 2003.
[7] R. Weber, H.-J. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High- Dimensional Spaces. In VLDB, 1998.
[8] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios. Spatial Outsourcing for Location-based Services. In ICDE, 2008.
[9] Brin, S.: Near neighbor search in large metric spaces. In VLDB pp. 574–584 (1995).
[10] Yuan, Y., Wang, G., Sun, Y.: Efficient peer-to-peer similarity query processing for high-dimensional data. In: Asia-Pacific Web Conference. pp. 195–201 (2010).
[11] Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L. Dynamic Authenticated Index Structures for Outsourced Databases. SIGMOD, 2006.
[12] Sion, R. Query Execution Assurance for Outsourced Databases. VLDB, 2005.
[13] Yang, Y., Papadopoulos, S., Papadias, D., Kollios, G. Spatial Outsourcing for Location-based Services. ICDE, 2008.
[14] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-Preserving Encryption for Numeric Data," in SIGMOD, 2004.

