Secure and Faster NN Queries on Outsourced Metric Data Assets

Renuka Bandi[#1], Madhu Babu Ch[*2]
[#1]M.Tech, CSE, BVRIT, Hyderabad, Andhra Pradesh, India
[#] Professor, Department of CSE, BVRIT, Hyderabad, Andhra Pradesh, India

Abstract--Cloud computing enables outsourcing of data in pay per use fashion. This will help data owners to have services of storage without the need for investment. However, they have security concerns as the cloud service providers focus is on storage instead of security. In this paper we consider a case where three parties are involved. They are cloud server, data owner and trusted client. Data owners outsource their metric data assets to cloud server. The cloud server is provides storage services to people across the world. The data owners can also give access to their data to trusted clients. We proposed techniques to have secure communication among the three parties. The data flown between the data owners and cloud server is protected as the data is transformed and encrypted before string in cloud server. When any client accesses data, it is decrypted automatically and thus there is secure communication among the three parties. We built a prototype application that shows the efficiency of our security mechanisms. The empirical results revealed that the proposed system supports efficient NN queries besides enabling secure communication among the three parties.

Index Terms –Cloud storage, security, NN queries, metric data, cloud service provider

I. INTRODUCTION

Metric data is the data that can have some sort of relationships among the fields. Such data is sensitive and generated by astronomy, bioinformatics, medicine and seismology. As such data is increasingly generated day by day processing it is not easy with systems containing limited resources. Cloud computing is the technology that allows outsourcing of such data for huge computational activities. The data once stored to cloud can be given access to other users or parties with secure mechanisms. This means that the data can be accessed by owners and also trusted clients who have secure access to the data of owners.

There are security concerns as the data is valuable to data owners. The data protection has to be given paramount importance. Moreover the cloud service providers give less importance to protection of data while much importance is given to the storage facilities. In order to secure such data viable and affordable solutions are required that guarantee the privacy of data. Metric data is sensitive because it is very important data that is required by organizations. For instance NASA gets such data continuously from its satellites. The data this organization gets is very valuable for scientific researches. It cannot afford it to be compromised. DNA data is another data for metric data assets. It is widely used in biological sciences. It contains sensitive information about humans. For this reason much importance is given to metric data assets and its security when outsourced to cloud. Applications use the data for processing. The systems have to ensure security when the data is at rest and also when it is in transit. Queries made to such data are to be protected in addition to the data protection.

In this paper we present various techniques for securing communication among the cloud server, data owner and trusted clients. Metric data assets are used to test the proposed system. We built an application with user friendly interface to demonstrate the efficiency of the proposed system. The remainder of this paper is structured as follows. Section II reviews relevant literature. Section III provides information about the architecture and security mechanisms of the proposed cloud storage system. Section IV provides experimental results while section V concludes the paper.

II. PRIOR WORK

Review of prior works which are related to this work is presented in this section. The topics reviewed include privacy and security, metric data, cloud data security and indexing NN search. In this paper algorithms are proposed for making NN searches faster. There are some disk based indexes that are very famous. They are R*tree [2] and X-tree [1]. Multi-dimensional objects make use of such indexes. DNS sequences or time series objects are best candidates for those indexes. With indexing in place, complex objects can be stored and retrieved easily. Metric data indexing has been around for some time as explored in [3] and [4]. Three indexing methods

were described by them. They are MVP – tree [6], M-tree [5] and VP – tree [7]. M-tree is very famous among all these data structures. A variant of M-tree is also explored in [8]. When M-tree is used for indexing, every entry in the index has an object known as anchor object. It also contains a point to child node, a covering radius and a pre-computed distance between the parent and child. For search metric data NN queries are usually made. "Best-first" is the algorithm which is widely used for NN queries [4], [9]. Minimum distance between the objects and query is computed in the search pace. Other approaches also exist to make such queries more efficiently. These are called hatching techniques [10], [11]. These techniques are useful in retrieving results from the metric data assets. However they do not provide accurate results. There are many efficient algorithms for NN queries. They are LSH (Locally Sensitive Hashing) [11], DBH (Distance Based Hashing) [10], and more on DBH is found in [12]. For instance it protection function is given as follows.

$$\mathcal{PJF}_{a_i,b_i}(p) = \frac{dist^2(p,a_i) + dist^2(a_i,b_i) - dist^2(p,b_i)}{2 \cdot dist(a_i,b_i)}.$$

DBH is very useful technique. However, it has two important limitations. It may result in empty datasets or unable to optimize query performance. This paper overcomes these limitations by suing hashing technique which is flexible that avoids empty results and improves query performance. The privacy and security of outsourcing can be traced back to the history of outsourcing. In [13] the first idea was presented. Later on in [14] encryption based solution is provided for the first time. Data owner applying encryption concept was introduced in [15]. Order – preserving function based solution was explored in [16].

The concepts in [17] and [18] are similar to our work. Spatial transformation and R-tree are used in the solutions. The encryption mechanism known as secure scalar encryption production is used. However it does not use indexing concept. For this reason it is not effective. In [19] K-anonymity is applied to applications where privacy is required. The proposed solution provides security to communications among the three partie3s such as could server, cloud data owner and trust client.

III. PROPOSED SECRUITY MECHANISMS

The proposed system architecture is presented in figure 1. It has mechanisms that can be used to secure storage and retrieval and query processes among the three parties. The three entities who are involved in the proposed system use the proposed secure mechanisms to protect data communications. The operations are illustrated in figure 1.

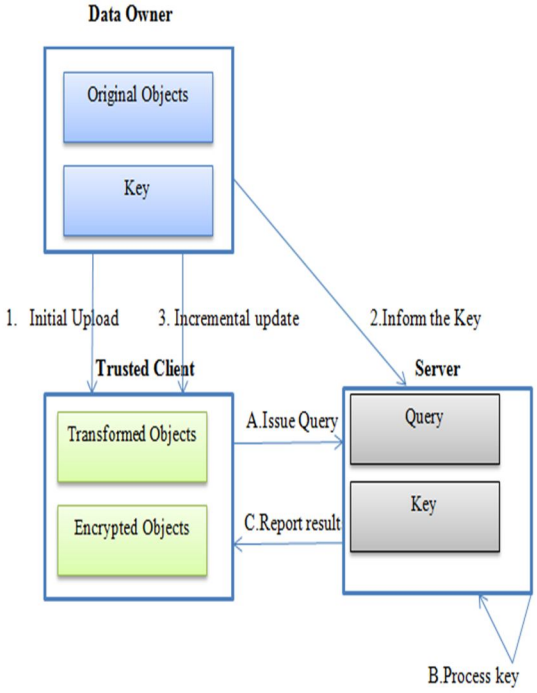

Fig. 1 –Overview of the proposed scenario

As seen in figure 1, the data owner uses the system to outsource his metric data to cloud server. However, this data storage takes place as per the given security mechanisms. Plain text is not saved to cloud server. Actually the data owner when he wants to outsource his data transforms it into encoded format and then encrypted before it is saved to cloud server. The server stores data and also indexes for fester processing of NN queries. Once data is uploaded, the cloud data owners can incrementally update it. Once data is stored in cloud server, the owners can provide privileges to trusted clients to access such data. The clients are known as trusted clients. The trusted clients perform queries on the storage server. The queries and also results are protected using secure algorithms proposed in this paper. Once results come to trusted client it is decrypted.

Security Mechanisms

In order to secure the metric data that is outsourced, there are three transformation functions proposed in this paper. These functions convert data into different format and without disturbing the data. The mechanisms used for data transformation are named as Flexible Distance Based Hashing (DBH), Encrypted Hierarchical Index Search (EHI) and Metric Preserving Transformation (MPT). First algorithm is meant for NN search process used by the client for making NN queries. In this case the cloud server makes use of index in order to retrieve data aster applying index in the search process. Thus perfect privacy is provided to the outsourced data.

Algorithm : EHI Searching Algorithm for Client.
Algorithm EHI-Search (Query object q, Encryption Key $\mathcal{CK}$, Integer λ)
1: request the server for the (encrypted) root node L_{root};
2: H:=new min-heap; p_{nn}:=NULL;
3: γ:=$\min_{e \in L_{root}} maxdist(q, e)$; ▷ derive NN distance bound
4: **for each** entry $e \in L_{root}$ such that $mindist(q, e) \leq \gamma$ **do**
5: insert the entry $\langle e, mindist(q, e) \rangle$ into H;
6: **while** H is not empty and its top entry's key $\leq \gamma$ **do**
7: pop next λ entries from H and insert them into a set S;
8: request the server for each (encrypted) child node of S;
9: **for each** retrieved node L_{cur} **do**
10: **if** L_{cur} is a leaf node **then** ▷ check for closer objects
11: update γ and p_{nn} by using objects in L_{cur};
12: **else** ▷ expand the entries of L_{cur}
13: γ:=$\min\{\gamma, \min_{e \in L_{cur}} maxdist(q, e)\}$;
14: **for each** $e \in L_{cur}$ such that $mindist(q, e) \leq \gamma$ **do**
15: insert the entry $\langle e, mindist(q, e) \rangle$ into H;
16: **return** p_{nn} as the result;

Listing 1 –EHI algorithm for searching

As can be seen in listing -1, the EHI algorithm is presented. This algorithm is meant for making NN search on the metric data present in cloud server. It has optimal data transfer cost while incurring large number of round trips to server. Listing 2 presents MPT building algorithm for data owner.

Algorithm : MPT Building Algorithm for Data Owner.
Algorithm MPT-Build (Data Set P, Encryption Key $\mathcal{CK}$, Integer A)
1: use a heuristic of Ref. [11] to select a set of A anchor objects from P;
2: Integer B:=$\lceil |P|/A \rceil$;
3: use a heuristic of Ref. [11] to assign each data object of P to an anchor object, subject to the capacity constraint B;
4: **for** i:=1 to A **do**
5: let a_i be the i-th anchor object;
6: let $a_i.S$ be the set of objects assigned to the anchor a_i;
7: r_i:=$\max_{p \in a_i.S} dist(a_i, p)$; ▷ compute covering radius
8: **for each** object $p \in a_i.S$ **do**
9: send the tuple $\langle p.id, \mathcal{OPE}(dist(a_i, p)), \mathcal{ECR}(p, \mathcal{CK}) \rangle$ to the server;

Listing 2- MPT building algorithm

As can be seen in listing 2, MPT building algorithm is presented. It is used by data owner while sending or outsourcing data to server. Listing 3 presents the FDH building algorithm for data owner.

Algorithm : FDH Building Algorithm for Data Owner.
Algorithm FDH-Build (Data Set P, Encryption Key $\mathcal{CK}$, Integer A)
1: **for** $i := 1$ to A **do** ▷ key generation
2: choose an object randomly from P as an anchor object a_i;
3: find the distance value r_i such that half of objects $p \in P$ satisfy $dist(a_i, p) \leq r_i$;
4: **for each** object $p \in P$ **do**
5: compute the encryption $\mathcal{ECR}(p, \mathcal{CK})$;
6: compute $\mathcal{BM}(p)$;
7: send the tuple $\langle p.id, \mathcal{BM}(p), \mathcal{ECR}(p, \mathcal{CK}) \rangle$ to the server;

Listing 3 –FDH Building Algorithm

The FDH building algorithm presented in listing 3 finishes the communication with the server with just a single round trip. However, it does not make any guarantee for accuracy of results.

IV. EXPERIEMNTAL RESULTS AND EVALUATION

The environment used for developing a prototype web application used by data owner and trusted client is a PC with 2GM of RAM and core 2 dual processor. The software used is Visual Studio with ASP.NET technology and Visual C# programming language. Datasets used areYEAST, MUSH, SHUTL and GFC. Experiments are made using the dataset with all security mechanisms described in the previous section for both data owner and also trusted client. The construction time and server CPU time (seconds) required by the three algorithms is provided in table 1.

Dataset	Construction Time			Server CPU Time		
	EHI	MPT	FDH	EHI	MPT	FDH
YEAST	0.016	0.094	0.313	0.001	0.001	0.049
MUSH	0.234	0.531	1.344	0.006	0.002	0.083
SHUTL	2.438	1.187	4.672	0.010	0.006	0.097
GFC	12.141	3.063	10.078	0.007	0.005	0.141

Table 1 –Construction and Server CPU Time for Three Transformations

As can be seen in table 1, the construction time and server CPU time are presented. For YEAST and MUSH datasets, EHI has good performance both for construction time and also CPU time in server. For SHUTL dataset MPT has good performance in terms of construction time and CPU time in server. In case of GFC dataset, also MPT has better performance. The evaluation of the algorithms with different datasets is visualized in the graphs presented in the following figures.

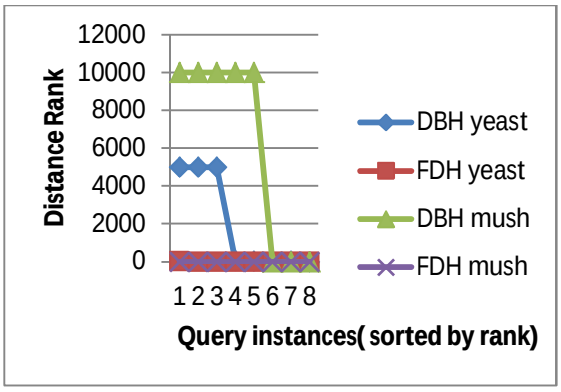

Fig. 2 –Rank of NN Search Results on YEAST and MUSH data

As can be seen in fig. 2, it is evident that the ranks of NN search results of DBH and FDB algorithms are presented for YEAST and MUSH datasets. On the datasets YEAST and MUSH the result rank of FDH is far better than that of DBH.

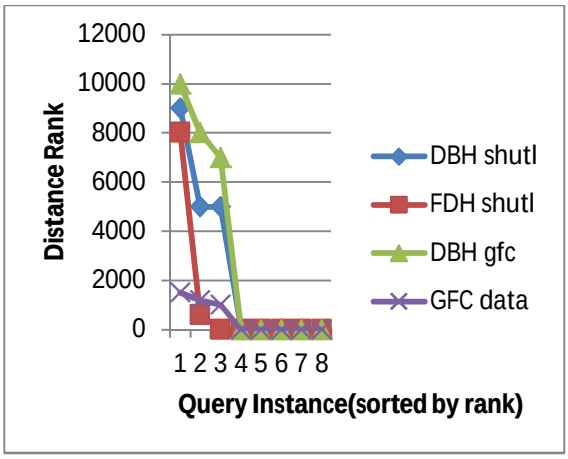

Fig. 3 – Rank of NN Search Results on SHUTL and GFC data

As can be seen in fig. 3, it is evident that the ranks of NN search results of DBH and FDB algorithms are presented for SHUTL and GFC datasets.

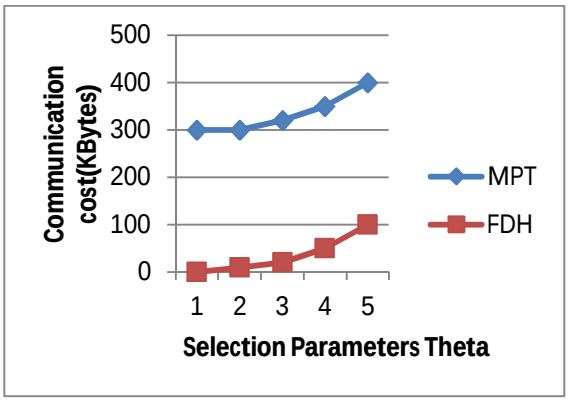

Fig. 4– Communication Cost
As can be seen in fig. 4 the communication cost of MPT and FDH algorithms is presented. As per the results shown the communication cost of FDH is far lesser than that of MPT.

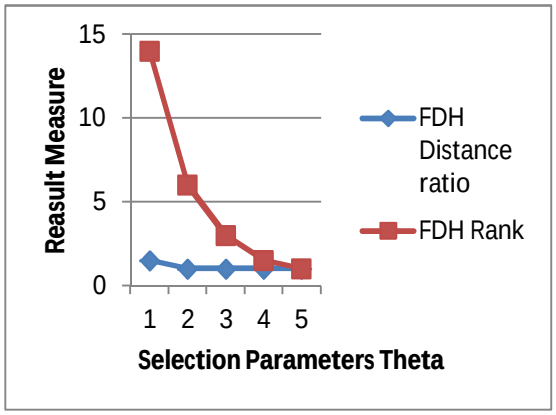

Fig. 5 –Result Measure
As can be seen in fig. 5, the result measure is presented with respect to FDH rank and FDH distance ratio.

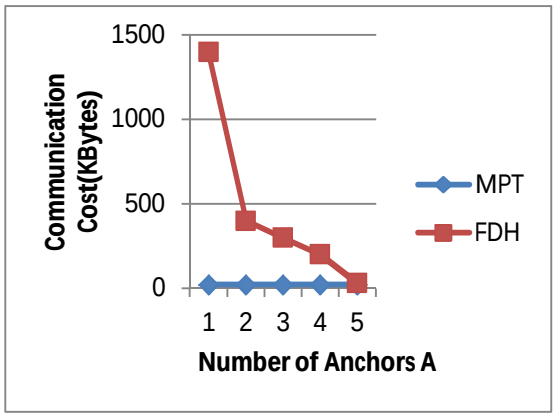

Fig. 6 – Communication Cost
As can be seen in fig. 6 the horizontal axis represents number of anchors while the vertical axis represents communication cost. They represent those details for both FDH and MPT algorithms. The MPT algorithm performance is far better that that of MPT with respect to number of anchors.

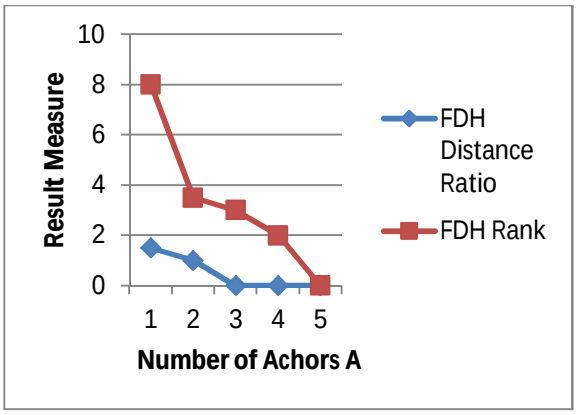

Fig. 7 – Result Measure
As can be seen in fig. 7, the result measure is presented with respect to FDH rank and FDH distance ratio with respect to number of objects.

V. CONCLUSION

In this paper we studied the NN search techniques on the outsourced metric data. The data is stored in cloud server. The data is stored by cloud data owner securely as it is transformed and encrypted before sent to server. The data considered include bioinformatics that is sensitive in nature. The existing solutions for NN search have tradeoff between data privacy and query efficiency. In this paper we designed search algorithms that help in both data privacy and query efficiency. Server is used to perform various search algorithms. MPT can store distance data in server. However, it needs two trips to reach to the server to process given task. However, the FDH algorithm reduces the round trips to single trip. However, this technique has not given guarantee of results' accuracy. Therefore in this paper all are utilized in order to provide fool proof security besides making the faster NN queries. The prototype application demonstrated the usefulness and the feasibility of the proposed approach.

REFERENCES

[1] S. Berchtold, D.A. Keim, and H.-P.Kriegel, "The X-Tree : An IndexStructure for High-Dimensional Data," Proc. 22nd Int'l Conf. VeryLarge Databases, pp. 28-39, 1996.

[2] N. Beckmann, H.-P.Kriegel, R. Schneider, and B. Seeger, "The R*-Tree: An Efficient and Robust Access Method for Points andRectangles," Proc. ACM SIGMOD Int'l Conf. Management of Data,pp. 322-331, 1990.

[3] E. Cha´vez, G. Navarro, R.A. Baeza-Yates, and J.L. Marroqui´n,"Searching in Metric Spaces," ACM Computing Surveys, vol. 33,no. 3, pp. 273-321, 2001.

[4] G.R. Hjaltason and H. Samet, "Index-Driven Similarity Search inMetric 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 AccessMethod for Similarity Search in Metric Spaces," Proc. Very LargeDatabases (VLDB), pp. 426-435, 1997.
[6] T. Bozkaya and Z.M. O ¨ zsoyoglu, "Indexing Large Metric Spacesfor Similarity Search Queries," ACM Trans. Database Systems,vol. 24, no. 3, pp. 361-404, 1999.
[7] P. Yianilos, "Data Structures and Algorithms for NearestNeighbor Search in General Metric Spaces," Proc. Fourth Ann.ACM-SIAM Symp.Discrete Algorithms (SODA), pp. 311-321, 1993.
[8] C.T. Jr, A.J.M. Traina, B. Seeger, and C. Faloutsos, "Slim-Trees:
High Performance Metric Trees Minimizing Overlap betweenNodes," Proc. Seventh Int'l Conf. Extending Database Technology(EDBT), pp. 51-65, 2000.
[9] T. Seidl and H.P. Kriegel, "Optimal Multi-Step k-NearestNeighbor Search," Proc. ACM SIGMOD Int'l Conf. Management ofData, pp. 154-165, 1998.
[10] V. Athitsos, M. Potamias, P. Papapetrou, and G. Kollios, "NearestNeighbor Retrieval Using Distance-Based Hashing," Proc. IEEE24th Int'l Conf. Data Eng. (ICDE), pp. 327-336, 2008.
[11] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in HighDimensions via Hashing," Proc. 25th Int'l Conf. Very LargeDatabases (VLDB), pp. 518-529, 1999.

[12] C. Faloutsos and K.-I. Lin, "FastMap: A Fast Algorithm forIndexing, Data-Mining and Visualization of Traditional andMultimedia Data Sets," Proc. ACM SIGMOD Int'l Conf. Managementof Data, pp. 163-174, 1995.
[13] H. Hacigu¨mu¨ s, S. Mehrotra, and B.R. Iyer, "Providing Database asa Service," Proc. 18th Int'l Conf. Data Eng. (ICDE), pp. 29-40, 2002.
[14] H. Hacigu¨mu¨ s, B.R. Iyer, C. Li, and S. Mehrotra, "Executing SQLover Encrypted Data in the Database-Service-Provider Model,"Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 216-227,2002.
[15] E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P.Samarati, "Balancing Confidentiality and Efficiency in UntrustedRelational DBMSs," Proc. 10th ACM Conf. Computer and Comm.Security (CCS), pp. 93-102, 2003.

[16] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-PreservingEncryption for Numeric Data," Proc. ACM SIGMOD Int'l Conf.Management of Data, pp. 563-574, 2004.
[17] M.L. Yiu, G. Ghinita, C.S. Jensen, and P. Kalnis, "OutsourcingSearch Services on Private Spatial Data," Proc. IEEE 25th Int'l Conf.Data Eng. (ICDE), pp. 1140-1143, 2009.
[18] W.K. Wong, D.W. Cheung, B. Kao, and N. Mamoulis, "Secure k-NN Computation on Encrypted Databases," Proc. 35th ACMSIGMOD Int'l Conf. Management of Data, pp. 139-152, 2009.
[19] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'lJ. Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5,pp. 557-570, 2002.

Authors

Renuka Bandi, She is pursuing M.Tech (CSE)in BVRIT , Hyderabad, AP, INDIA. She has received B.Tech Degree in Computer Science and Engineering. Her main research interest includes Cloud Computing and Data mining.

Madhu Babu Chunduri. He is currently with the Department of Computer Science and Engineering, BVRIT, Andhra Pradesh, India. Having 15 years of teaching experience**.** His main research interest includes Software Engineering and Data Mining.